

A Cut-free Sequent Calculus for Bi-Intuitionistic Logic: Extended Version

Linda Buisman and Rajeev Goré

¹ The Australian National University
Canberra ACT 0200, Australia

² Logic and Computation Programme
Canberra Research Laboratory, NICTA*, Australia
{Linda.Buisman|Rajeev.Gore}@anu.edu.au

Abstract. Bi-intuitionistic logic is the extension of intuitionistic logic with a connective dual to implication. Bi-intuitionistic logic was introduced by Rauszer as a Hilbert calculus with algebraic and Kripke semantics. But her subsequent “cut-free” sequent calculus for **BiInt** has recently been shown by Uustalu to fail cut-elimination. We present a new cut-free sequent calculus for **BiInt**, and prove it sound and complete with respect to its Kripke semantics. Ensuring completeness is complicated by the interaction between implication and its dual, similarly to future and past modalities in tense logic. Our calculus handles this interaction using extended sequents which pass information from premises to conclusions using variables instantiated at the leaves of failed derivation trees. Our simple termination argument allows our calculus to be used for automated deduction, although this is not its main purpose.

1 Introduction

Propositional intuitionistic logic (**Int**) has connectives \rightarrow , \wedge , \vee and \neg , with $\neg\varphi$ often defined as $\neg\varphi := \varphi \rightarrow \perp$. **Int** has a well-known Kripke semantics, where a possible world w makes $\varphi \rightarrow \psi$ true if every successor v that makes φ true also makes ψ true. **Int** also has an algebraic semantics in terms of Heyting algebras, and there is a well-known embedding from **Int** into the classical modal logic **S4**. **Int** is constructive in that it rejects the Law of Excluded Middle: that is, $\varphi \vee \neg\varphi$ is not a theorem of **Int**.

Propositional dual intuitionistic logic (**DualInt**) has connectives \multimap , \wedge , \vee and \sim , with $\sim\varphi$ often defined as $\sim\varphi := \top \multimap \varphi$. **DualInt** also has Kripke semantics, where a possible world w makes $\varphi \multimap \psi$ true if there exists a predecessor v where φ holds, but ψ does not hold: that is, φ *excludes* ψ . Thus, the \multimap connective of **DualInt** is dual to implication in **Int**. **DualInt** also has algebraic

* National ICT Australia is funded by the Australian Government’s Dept of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia’s Ability and the ICT Centre of Excellence program.

semantics in terms of Brouwer algebras [13]. There is a less well-known embedding from **DualInt** into **S4**. **DualInt** is para-consistent in that it rejects the Law of Non-contradiction: that is, $\varphi \wedge \sim \varphi$ is **DualInt**-satisfiable. Various names have been used for \multimap : coimplication [24, 23], subtraction [2, 3], pseudo-difference [16], explication [15]. We refer to it as exclusion.

Bi-intuitionistic logic (**BiInt**), also known as subtractive logic and Heyting-Brouwer logic, is the union of **Int** and **DualInt**, and it is a conservative extension of both. **BiInt** was first studied by Rauszer [15, 16]. **BiInt** is an interesting logic to study, since it combines the constructive aspects of **Int** with the para-consistency of **DualInt**. While every **Int**-theorem is also a **BiInt**-theorem, adding **DualInt** connectives introduces a non-constructive aspect to the logic – the disjunction property does not hold for **BiInt** formulae if they contain \multimap . Note that **BiInt** differs from intuitionistic logic with constructive negation, also known as constructible falsity [14], where the disjunction property does hold.

While the proof theory of **Int** and **DualInt** separately has been studied extensively and there are many cut-free sequent systems for **Int** (for example, [8, 6, 5]) and **DualInt** (for example, [20, 4]), the case for **BiInt** is less satisfactory. Although Rauszer presented a sequent calculus for **BiInt** in [15] and “proved” it cut-free, Uustalu has recently given a counter-example [21] to her cut-elimination theorem: the formula $p \rightarrow (q \vee (r \rightarrow ((p \multimap q) \wedge r)))$ is **BiInt**-valid, but cannot be derived in Rauszer’s calculus without the cut rule. Similarly, Uustalu’s counterexample shows that Crolard’s sequent calculus [2] for **BiInt** is not cut-free. Uustalu’s counterexample fails in both Rauszer’s and Crolard’s calculi because they limit certain sequent rules to singleton succedents or antecedents in the conclusion, and the rules do not capture the interaction between implication and exclusion.

Uustalu and Pinto have also given a cut-free sequent-calculus for **BiInt** in [23]. Since only the abstract of this work has been published so far, we have not been able to examine their sequent rules, or verify their proofs. According to the abstract [23] and personal communication with Uustalu [22], his calculus uses labelled formulae, thereby utilising some semantic aspects, such as explicit worlds and accessibility, directly in the rules. Hence a traditional cut-free sequent calculus for **BiInt** is still an open problem.

We present a new purely syntactic cut-free sequent calculus for **BiInt**. We avoid Rauszer’s and Crolard’s restrictions on the antecedents and succedents for certain rules by basing our rules on Dragalin’s **GHPC** [5] which allows multiple formulae on both sides of sequents. To maintain intuitionistic soundness, we restrict the *premise* of the implication-right rule to a singleton in the succedent. Dually, the premise of our exclusion-left rule is restricted to a singleton in the antecedent. But using Dragalin’s calculus and its dual does not give us **BiInt** completeness. We therefore follow Schwendimann [17], and use sequents which pass relevant information from premises to conclusions using variables instantiated at the leaves of failed derivation trees. We then recompute parts of our derivation trees using the new information, similarly to the restart technique of [11]. Our calculus thus uses a purely syntactic addition to traditional sequents,

rather than resorting to a semantic mechanism such as labels. Our termination argument also relies on two new rules from Švejdar [18].

If we were interested only in decision procedures, we could obtain a decision procedure for **BiInt** by embedding it into the tense logic **Kt.S4** [24], and using tableaux for description logics with inverse roles [11]. However, an embedding into **Kt.S4** provides no proof-theoretic insights into **BiInt** itself. Moreover, the restart technique of Horrocks et al. [11] involves non-deterministic expansion of disjunctions, which is complicated by inverse roles. Their actual implementation avoids this non-determinism by keeping a global view of the whole counter-model under construction. In contrast, we handle this non-determinism by syntactically encoding it using variables and extended formulae, neither of which have a semantic content. Our purely syntactic approach is preferable for proof-theoretic reasons, since models are never explicitly involved in the proof system: see Remark 3.

The rest of the paper is organized as follows. In Section 2, we define the syntax and semantics of **BiInt**. In Section 3, we introduce our sequent calculus **GBiInt** and give an example derivation of Uustalu’s interaction formula. We prove the soundness and completeness of **GBiInt** in Sections 4 and 5 respectively. In Section 6, we outline further work.

2 Syntax and Semantics of BiInt

In this section we introduce the syntax and semantics of **BiInt**.

Definition 1 (Syntax). *The formulae of BiInt are defined as:*

$$p ::= \top \mid \perp \mid p_0 \mid p_1 \mid \dots \quad (2.1)$$

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \prec \varphi \mid \neg\varphi \mid \sim\varphi \quad (2.2)$$

We refer to the set of atoms as *Atoms*, and we refer to the set of **BiInt** formulae as *Fml*.

The connectives \neg and \rightarrow are those of intuitionistic logic, and the connectives \sim and \prec are those of dual intuitionistic logic. The connectives \vee and \wedge are from both.

Definition 2 (Length). *The length of a BiInt formula χ is defined as:*

$$\text{len}(\chi) = \begin{cases} 1 & \text{if } \chi \in \text{Atoms} \\ \text{len}(\varphi) + 1 & \text{if } \chi \in \{\neg\varphi, \sim\varphi\} \\ \text{len}(\varphi) + \text{len}(\psi) + 1 & \text{if } \chi \in \{\varphi \vee \psi, \varphi \wedge \psi, \varphi \rightarrow \psi, \varphi \prec \psi\}. \end{cases}$$

We use the language of classical first-order logic when reasoning about **BiInt** at the meta-level.

Definition 3 (Frame). *A BiInt frame is a pair $\langle \mathcal{W}, \mathcal{R} \rangle$, where:*

1. \mathcal{W} is a non-empty set of worlds;
2. $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$ is the binary accessibility relation;
3. \mathcal{R} is reflexive, i.e., $\forall u \in \mathcal{W}. u\mathcal{R}u$;
4. \mathcal{R} is transitive, i.e., $\forall u, v, w \in \mathcal{W}. (u\mathcal{R}v \ \& \ v\mathcal{R}w \Rightarrow u\mathcal{R}w)$.

Definition 4 (Model). A BiInt model is a triple $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, where:

1. $\langle \mathcal{W}, \mathcal{R} \rangle$ is a BiInt frame;
2. The truth valuation ϑ is a function $\mathcal{W} \times \text{Atoms} \rightarrow \{\text{true}, \text{false}\}$, which tells us the truth value of an atom at a world;
3. The persistence property holds:
 $\forall u, w \in \mathcal{W}. \forall p \in \text{Atoms}. (\vartheta(w, p) = \text{true} \ \& \ w\mathcal{R}u) \Rightarrow (\vartheta(u, p) = \text{true})$;
4. $\forall w \in \mathcal{W}. \vartheta(w, \top) = \text{true}$;
5. $\forall w \in \mathcal{W}. \vartheta(w, \perp) = \text{false}$.

Definition 5 (Forcing of atoms). Given a model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, a world $w \in \mathcal{W}$ and an atom $p \in \text{Atoms}$, we write $w \models p$ if $\vartheta(w, p) = \text{true}$. We pronounce \models as “forces”, and we pronounce $\not\models$ as “rejects”.

Definition 6 (Forcing of formulae). Given a model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, a world $w \in \mathcal{W}$ and formulae $\varphi, \psi \in \text{Fml}$, we write:

$$\begin{array}{ll}
w \models \varphi \vee \psi & \text{if } w \models \varphi \text{ or } w \models \psi \\
w \models \varphi \wedge \psi & \text{if } w \models \varphi \ \& \ w \models \psi \\
w \models \neg \varphi & \text{if } \forall u \in \mathcal{W}. [w\mathcal{R}u \Rightarrow (u \not\models \varphi)] \\
w \models \varphi \rightarrow \psi & \text{if } \forall u \in \mathcal{W}. [w\mathcal{R}u \Rightarrow (u \not\models \varphi \text{ or } u \models \psi)] \\
w \models \sim \varphi & \text{if } \exists u \in \mathcal{W}. [u\mathcal{R}w \ \& \ u \not\models \varphi] \\
w \models \varphi \prec \psi & \text{if } \exists u \in \mathcal{W}. [u\mathcal{R}w \ \& \ u \models \varphi \ \& \ u \not\models \psi]
\end{array}$$

From the semantics, it can be seen that the connectives \neg and \sim can be derived from \rightarrow and \prec respectively. Therefore from now on we restrict our attention to the connectives \rightarrow , \prec , \wedge , \vee only.

Lemma 1. The persistence property also holds for formulae, that is:

$$\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle. \forall u, w \in \mathcal{W}. \forall \varphi \in \text{Fml}. (w \models \varphi \ \& \ w\mathcal{R}u \Rightarrow u \models \varphi).$$

Proof. By induction on the length of φ .

Lemma 2. The reverse persistence property holds:

$$\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle. \forall u, w \in \mathcal{W}. \forall \varphi \in \text{Fml}. (w \not\models \varphi \ \& \ u\mathcal{R}w \Rightarrow u \not\models \varphi).$$

Proof. Reverse persistence follows from persistence, because the truth valuation is binary. That is, suppose for a contradiction that

$$\exists \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle, \exists u, w \in \mathcal{W}. \exists \varphi \in \text{Fml}. (w \not\models \varphi \ \& \ u\mathcal{R}w \ \& \ u \models \varphi).$$

Then $u \models \varphi$ and $u\mathcal{R}w$ together with the persistence property give us $w \models \varphi$, which contradicts $w \not\models \varphi$.

We write ϵ to mean the empty set. Given two sets of formulae Δ and Γ , we write Δ, Γ for $\Delta \cup \Gamma$. Given a set of formulae Δ and a formula φ , we write Δ, φ for $\Delta \cup \{\varphi\}$.

Definition 7. Given a model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, a world $w \in \mathcal{W}$ and sets of formulae Γ and Δ , we write:

$$\begin{aligned} w \models \Gamma & \text{ if } \forall \varphi \in \Gamma. w \models \varphi \\ w \models \Delta & \text{ if } \forall \varphi \in \Delta. w \models \varphi. \end{aligned}$$

As a corollary, for any world w , we vacuously have $w \models \epsilon$ and $w \models \epsilon$.

Definition 8 (Consequence). Given two sets Γ and Δ of formulae, $\Gamma \Vdash_{\text{BiInt}} \Delta$ means:

$$\forall \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle. \forall w \in \mathcal{W}. \text{ if } w \models \Gamma \text{ then } \exists \varphi \in \Delta. w \models \varphi.$$

We write $\Gamma \nVdash_{\text{BiInt}} \Delta$ to mean that it is not the case that $\Gamma \Vdash_{\text{BiInt}} \Delta$, that is:

$$\exists \mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle. \exists w \in \mathcal{W}. (w \models \Gamma \ \& \ w \not\models \Delta).$$

Thus $\Gamma \nVdash_{\text{BiInt}} \Delta$ means that $\Gamma \Vdash_{\text{BiInt}} \Delta$ is falsifiable.

We wish to prove $\Gamma \Vdash_{\text{BiInt}} \Delta$ by failing to falsify $\Gamma \nVdash_{\text{BiInt}} \Delta$. By Definition 8, $\Gamma \nVdash_{\text{BiInt}} \Delta$ means that there exists a **BiInt** model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ that contains a world $w_0 \in \mathcal{W}$ such that $w_0 \models \Gamma$ and $w_0 \not\models \Delta$. We therefore try to construct the model using a standard *counter-model construction* approach: see [7]. We shall start with an initial world w_0 and assume that $w_0 \models \Gamma$ and $w_0 \not\models \Delta$, and then systematically decompose the formulae in Γ and Δ . The procedure will either:

- lead to a contradiction and therefore conclude that it cannot be the case that $w_0 \models \Gamma$ and $w_0 \not\models \Delta$, therefore $\Gamma \Vdash_{\text{BiInt}} \Delta$ holds, OR
- construct the counter-model successfully and therefore demonstrate that it is possible that $w_0 \models \Gamma$ and $w_0 \not\models \Delta$, therefore $\Gamma \nVdash_{\text{BiInt}} \Delta$ does not hold.

3 Our Sequent Calculus GBiInt

We now present **GBiInt**, a Gentzen-style sequent calculus for **BiInt**. The sequents have a non-traditional component in the form of variables that are instantiated at the leaves of the derivation tree, and passed back to lower sequents from premises to conclusion. Note that the variables are not names for Kripke models and have no semantic content.

3.1 Sequents

First, we introduce an extended syntax that will help us in the presentation of some of our sequent rules.

Definition 9 (Extended Syntax). The extended **BiInt** formulae are defined as follows:

1. If φ is a **BiInt** formula, then φ is an extended **BiInt** formula,
2. If \mathcal{S} and \mathcal{P} are sets of sets of **BiInt** formulae, then $\bigvee \mathcal{S}$ and $\bigwedge \mathcal{P}$ are extended **BiInt** formulae.

If $\mathcal{S} = \{\{\varphi_0^0, \dots, \varphi_0^n\}, \dots, \{\varphi_m^0, \dots, \varphi_m^k\}\}$ and $\mathcal{P} = \{\{\psi_0^0, \dots, \psi_0^n\}, \dots, \{\psi_m^0, \dots, \psi_m^k\}\}$, then from every extended **BiInt** formula we can obtain a **BiInt** formula as follows:

$$\begin{aligned}\bigvee \mathcal{S} &\equiv (\varphi_0^0 \wedge \dots \wedge \varphi_0^n) \vee \dots \vee (\varphi_m^0 \wedge \dots \wedge \varphi_m^k) \\ \bigwedge \mathcal{P} &\equiv (\psi_0^0 \vee \dots \vee \psi_0^n) \wedge \dots \wedge (\psi_m^0 \vee \dots \vee \psi_m^k).\end{aligned}$$

From now on, we implicitly treat extended **BiInt** formulae as their **BiInt** equivalents. The following semantics follows directly from Definition 9:

Definition 10 (Semantics of Extended Syntax). Given a **BiInt** model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$, and a world $w_0 \in \mathcal{W}$, we write:

$$\begin{aligned}w &\models \bigvee \mathcal{S} \quad \text{if} \quad \exists \Gamma \in \mathcal{S}. w \models \Gamma \\ w &\models \bigwedge \mathcal{P} \quad \text{if} \quad \exists \Delta \in \mathcal{P}. w \models \Delta.\end{aligned}$$

We can now extend the definition of forcing and rejecting to extended **BiInt** formulae in the obvious way. If Γ and Δ are sets of extended **BiInt** formulae viewed as their **BiInt** equivalents, and φ is an extended **BiInt** formula viewed as its **BiInt** equivalent, then:

$$\begin{aligned}w &\models \Gamma \quad \text{if} \quad \forall \varphi \in \Gamma. w \models \varphi \\ w &\models \Delta \quad \text{if} \quad \forall \varphi \in \Delta. w \models \varphi.\end{aligned}$$

Definition 11 (Sequent). A **GBiInt** sequent is an expression of the form

$$\frac{\mathcal{S}}{\mathcal{P}} \parallel \Gamma \vdash \Delta$$

and consists of the following components:

Left hand side (LHS): Γ , a set of extended **BiInt** formulae;

Right hand side (RHS): Δ , a set of extended **BiInt** formulae;

Variables: \mathcal{S}, \mathcal{P} , each of which is a set of sets of formulae.

We shall sometimes use $\Gamma \vdash \Delta$ to refer to sequents, ignoring the variable values for readability. We shall only do that in cases where the values of the variables are not important to the discussion. Note that the variables do not contain extended **BiInt** formulae.

We now define the meaning of a sequent in terms of the counter-model under construction.

Definition 12 (Falsifiability). A sequent

$$\frac{\mathcal{S}}{\mathcal{P}} \parallel \Gamma \vdash \Delta$$

is falsifiable [at w_0 in \mathcal{M}] if and only if there exists a **BiInt** model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and $\exists w_0 \in \mathcal{W}$ such that $w_0 \models \Gamma$ and $w_0 \not\models \Delta$.

Definition 13 (Variable conditions). We say the variable conditions of a sequent

$$\gamma = \frac{\mathcal{S}}{\mathcal{P}} \parallel \Gamma \vdash \Delta$$

hold if and only if γ is falsifiable at w_0 in some model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ and the following conditions hold:

S-condition: Successor condition

$$\exists \Sigma \in \mathcal{S}. \forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \models \Sigma$$

P-condition: Predecessor condition

$$\exists \Pi \in \mathcal{P}. \forall w \in \mathcal{W}. w \mathcal{R} w_0 \Rightarrow w \models \Pi$$

Lemma 3. A sequent $\Gamma \vdash \Delta$ is not falsifiable if and only if $\Gamma \Vdash_{\text{BiInt}} \Delta$.

Proof. Applying the negation of Definition 12 to $\Gamma \vdash \Delta$ gives $\Gamma \Vdash_{\text{BiInt}} \Delta$.

3.2 Sequent Rules

Definition 14 (Sequent Rule). A sequent rule is of one of the forms

$$\begin{array}{ccc} \text{(name)} \frac{\gamma_1 \cdots \gamma_n}{\gamma_0} & & \text{(name)} \frac{\gamma_1 \cdots \gamma_n}{\gamma_0} \\ \text{side conditions} & & \text{side conditions} \end{array}$$

where γ_i , $0 \leq i \leq n$ for $n \geq 0$, are sequents. The rule consists of the following components:

Conclusion: γ_0 , written below the horizontal line;

Premise(s): Optional, $\gamma_1, \dots, \gamma_n$, written above the horizontal line;

Name: Written to the left of the horizontal line;

Side conditions: Optional, written underneath the rule;

Branching: Universal (indicated by a solid line) or existential (indicated by a dashed line); explained shortly.

To achieve completeness and termination for **BiInt**, we combine a number of ideas from various existing systems for **Int**, as well as use variables for updating worlds with relevant information received from successors and predecessors. Our rules can be divided into two groups: traditional (Fig. 1) and non-traditional (Fig. 2).

Our **traditional** rules (Fig. 1) are based on Dragalin's GHPC [5] for **Int** because we require multiple formulae in the succedents and antecedents of sequents for completeness; we have added symmetric rules for the **DualInt** connective \neg . The main difference is that our (\rightarrow_L) rule and the symmetric (\neg_R) carry their principal formula and all side formulae into the premises. Our rules for \wedge and \vee also carry their principal formula into their premises to assist with termination. Note that there are other approaches to a terminating sequent calculus for **Int**, e.g., Dyckhoff's contraction-free calculi [6], or history methods by Heuerding et al. [10] and Howe [12]. These methods are less suitable when the interaction

$$\begin{array}{c}
(Id) \frac{}{\mathcal{S}:=\epsilon \parallel \Gamma, \varphi \vdash \Delta, \varphi} \quad (\perp_L) \frac{}{\mathcal{S}:=\epsilon \parallel \Gamma, \perp \vdash \Delta} \quad (\top_R) \frac{}{\mathcal{S}:=\epsilon \parallel \Gamma \vdash \Delta, \top} \\
\\
(\wedge_L) \frac{\mathcal{S}_1 \parallel \Gamma, \varphi \wedge \psi, \varphi, \psi \vdash \Delta}{\mathcal{S}:=\mathcal{S}_1 \parallel \Gamma, \varphi \wedge \psi \vdash \Delta} \quad (\wedge_R) \frac{\mathcal{S}_1 \parallel \Gamma \vdash \Delta, \varphi \wedge \psi, \varphi \quad \mathcal{S}_2 \parallel \Gamma \vdash \Delta, \varphi \wedge \psi, \psi}{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2 \parallel \Gamma \vdash \Delta, \varphi \wedge \psi} \\
\\
(\vee_R) \frac{\mathcal{S}_1 \parallel \Gamma \vdash \Delta, \varphi \vee \psi, \varphi, \psi}{\mathcal{S}:=\mathcal{S}_1 \parallel \Gamma \vdash \Delta, \varphi \vee \psi} \quad (\vee_L) \frac{\mathcal{S}_1 \parallel \Gamma, \varphi \vee \psi, \varphi \vdash \Delta \quad \mathcal{S}_2 \parallel \Gamma, \varphi \vee \psi, \psi \vdash \Delta}{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2 \parallel \Gamma, \varphi \vee \psi \vdash \Delta} \\
\\
(\rightarrow_L) \frac{\mathcal{S}_1 \parallel \Gamma, \varphi \rightarrow \psi \vdash \varphi, \Delta \quad \mathcal{S}_2 \parallel \Gamma, \varphi \rightarrow \psi, \psi \vdash \Delta}{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2 \parallel \Gamma, \varphi \rightarrow \psi \vdash \Delta} \\
\\
(\neg<_R) \frac{\mathcal{S}_1 \parallel \Gamma, \psi \vdash \Delta, \varphi \neg< \psi \quad \mathcal{S}_2 \parallel \Gamma \vdash \Delta, \varphi \neg< \psi, \varphi}{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2 \parallel \Gamma \vdash \Delta, \varphi \neg< \psi}
\end{array}$$

For every rule with premises π_i and conclusion γ , apply the rule only if:
 $\forall \pi_i. (LHS_{\pi_i} \not\subseteq LHS_{\gamma} \text{ or } RHS_{\pi_i} \not\subseteq RHS_{\gamma})$

Fig. 1. GBInt rules - traditional

between **Int** and **DualInt** formulae needs to be considered, since they erase potentially relevant formulae too soon during backward proof search. Moreover, we found it easier to prove semantic completeness with our loop-checking method than with history-based methods since both [10] and [12] prove completeness using syntactic transformations of derivations. Consequently, while **GBInt** is sound and complete for the **Int** (and **DualInt**) fragment of **BiInt**, it is unlikely to be as efficient on the fragment as these specific calculi.

Our rules for \rightarrow on the right and $\neg<$ on the left (Fig. 2) are **non-traditional**. The (\rightarrow_R) and $(\neg<_L)$ rules have *two* premises instead of one, and they are connected by **existential branching** as indicated by the dotted horizontal line. Existential branching means that the conclusion is derivable if *some* premise is derivable; thus it is dual to the conventional universal branching, where the conclusion is derivable if *all* premises are derivable. We chose existential branching rather than two separate non-invertible rules so the left premise can communicate information via variables to the right premise. This inter-premise communication and the use of variables is crucial to proving interaction formulae of **BiInt**, and it gives our calculus an **operational reading**.

When applying an existential branching rule during backward proof search, we first create the left premise. If the left premise is non-derivable, then it returns the variables \mathcal{S}_1 and \mathcal{P}_1 . We then use these variables to create the right premise, which corresponds to the same world as the conclusion, but with updated infor-

$$\begin{array}{c}
(Ret) \frac{}{\frac{\mathcal{S} := \{\Gamma\}}{\mathcal{P} := \{\Delta\}} \parallel \Gamma \vdash \Delta} \\
\text{where no other rule is applicable} \\
\\
(\rightarrow_R) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi, \psi}{\frac{\mathcal{S} := \mathcal{S}_1}{\mathcal{P} := \mathcal{P}_1} \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi} \qquad
(-<_L) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma, \varphi, \varphi -< \psi \vdash \Delta}{\frac{\mathcal{S} := \mathcal{S}_1}{\mathcal{P} := \mathcal{P}_1} \parallel \Gamma, \varphi -< \psi \vdash \Delta} \\
\\
(\rightarrow_R) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma, \varphi \vdash \psi \quad \frac{\mathcal{S}_2}{\mathcal{P}_2} \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi, \wedge \mathcal{P}_1}{\frac{\mathcal{S}/\mathcal{P} := \begin{cases} \mathcal{S}_1/\mathcal{P}_1 & \text{if } \mathcal{P}_1 = \epsilon \\ \mathcal{S}_2/\mathcal{P}_2 & \text{if right prem created} \\ \{\Gamma\}/\{\Delta, \varphi \rightarrow \psi\} & \text{otherwise} \end{cases}}{\text{right prem created only if } \mathcal{P}_1 \neq \epsilon \ \& \ \forall \Pi_i \in \mathcal{P}_1. \Pi_i \not\subseteq \{\Delta, \varphi \rightarrow \psi\}}} \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi \\
\\
(-<_L) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \varphi \vdash \Delta, \psi \quad \frac{\mathcal{S}_2}{\mathcal{P}_2} \parallel \Gamma, \varphi -< \psi, \vee \mathcal{S}_1 \vdash \Delta}{\frac{\mathcal{S}/\mathcal{P} := \begin{cases} \mathcal{S}_1/\mathcal{P}_1 & \text{if } \mathcal{S}_1 = \epsilon \\ \mathcal{S}_2/\mathcal{P}_2 & \text{if right prem created} \\ \{\Gamma, \varphi -< \psi\}/\{\Delta\} & \text{otherwise} \end{cases}}{\text{right prem created only if } \mathcal{S}_1 \neq \epsilon \ \& \ \forall \Sigma_i \in \mathcal{S}_1. \Sigma_i \not\subseteq \{\Gamma, \varphi -< \psi\}}} \parallel \Gamma, \varphi -< \psi \vdash \Delta \\
\\
(\wedge_R) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma \vdash \Delta, \Pi_1 \quad \dots \quad \frac{\mathcal{S}_n}{\mathcal{P}_n} \parallel \Gamma \vdash \Delta, \Pi_n}{\frac{\mathcal{S} := \bigcup_{i=1}^n \mathcal{S}_i}{\mathcal{P} := \bigcup_{i=1}^n \mathcal{P}_i} \parallel \Gamma \vdash \Delta, \wedge \Pi} \qquad
(\vee_L) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma, \Sigma_1 \vdash \Delta \quad \dots \quad \frac{\mathcal{S}_n}{\mathcal{P}_n} \parallel \Gamma, \Sigma_n \vdash \Delta}{\frac{\mathcal{S} := \bigcup_{i=1}^n \mathcal{S}_i}{\mathcal{P} := \bigcup_{i=1}^n \mathcal{P}_i} \parallel \Gamma, \vee \Sigma \vdash \Delta}
\end{array}$$

For every universally branching rule with premises π_i and conclusion γ ,
 apply the rule only if: $\forall \pi_i. (LHS_{\pi_i} \not\subseteq LHS_{\gamma} \text{ or } RHS_{\pi_i} \not\subseteq RHS_{\gamma})$

For every existentially branching rule with left premise π and conclusion γ ,
 apply the rule only if: $LHS_{\pi} \not\subseteq LHS_{\gamma} \text{ or } RHS_{\pi} \not\subseteq RHS_{\gamma}$

Fig. 2. GBiInt rules - non-traditional

mation. Our existential branching rules work together with (Ret) , which assigns the variables at non-derivable leaves of failed derivation trees, and (\wedge_R) and (\vee_L) , which extract the different variable choices at existential branching rules.

The conclusion of each of our rules **assigns the variables** based on the variables returned from the premise(s), and we use the indices $i, 1, 2$ to indicate the premise from which the variable takes its value. For rules with a single premise, the variables are simply passed down from premise to conclusion. For example, the conclusion of (\wedge_L) in Fig. 1 assigns $\mathcal{S} := \mathcal{S}_1$, where \mathcal{S}_1 is the value of the variable at the premise. However, for rules with multiple universally branching premises, we take a union of the sets of sets corresponding to each falsifiable premise. For example, the conclusion of (\wedge_R) in Fig. 2 assigns $\mathcal{S} := \bigcup_{i=1}^n \mathcal{S}_i$, where \mathcal{S}_i is the value of the variable at the i -th premise.

This way, the sets of sets stored in our variables **determinise** the return of formulae to lower sequents – each non-derivable premise corresponds to an open branch, and at this point we do not know whether it will stay open once

processed in conjunction with lower sequents. Therefore, we need to temporarily keep all open branches: see Example 2. Then the intuition behind adding $\bigwedge \mathcal{P}$ to the right premise of (\rightarrow_R) is that the subsequent application of (\bigwedge_R) will create one or more premises, depending on the cardinality of \mathcal{P} . Since \mathcal{P} is a set of sets representing all the open branches, all of the premises of (\bigwedge_R) have to be derivable in order to obtain a derivation. On the other hand, if some premises of (\bigwedge_R) are non-derivable (open), we form the set that consists of the union of the variables returned by those premises, and pass the union back to lower sequents, and so on. The premises that are derivable contribute only ϵ and are thus ignored by the union operator. Also, we only create the right premise of (\rightarrow_R) if **every** member of \mathcal{P} introduces new formulae to the current world. Otherwise, the current world already contains one of the open branches, which would still remain open after an application of (\bigwedge_R) . To summarise, the sets-of-sets concept of variables is critical to the soundness of **GBiInt**, as it allows us to remember the required choices arising further up the tree.

The **extended syntax** allows us to syntactically encode the variable choices described above. While the variables \mathcal{S} and \mathcal{P} are sets of sets when we pass them down the tree and combine them using set union, we use $\bigvee \mathcal{S}$ on the left and $\bigwedge \mathcal{P}$ on the right of the sequent to reflect these choices when we add $\bigvee \mathcal{S}$ or $\bigwedge \mathcal{P}$ to the right premise of an existentially branching rule. Then the (\bigvee_L) and (\bigwedge_R) rules break down the extended formulae $\bigvee \mathcal{S}$ and $\bigwedge \mathcal{P}$ to yield several premises, each corresponding to one variable choice. Thus the extended syntax allows us to give an intuitive syntactic representation of the variable choices.

We have also added the rule (\rightarrow_R^I) for implication on the right (and dually, $(\neg\leftarrow_L^I)$) originally given by Švejdar [18]. Rather than immediately creating the successor for a rejected $\varphi \rightarrow \psi$, the (\rightarrow_R^I) rule first pre-emptively adds ψ to the right hand side of the sequent. Although Švejdar himself does not give the semantics behind this rule, and is unable to explain the precise role it plays in his calculus, it is very useful in our termination proof. The rule effectively uses the reverse persistence property – if some successor v forces φ and rejects ψ , then the current world w must reject ψ too, for if w forces ψ , then by forward persistence so does v , thus giving a contradiction.

The **side condition** on each of our rules is a general **blocking** condition, where we only explore the premise(s), if they are different from the conclusion. For example, in the (\bigwedge_R) case, the blocking condition means that we apply the rule in backward proof search only if $\varphi \notin \Delta$ and $\psi \notin \Delta$, since otherwise some premise would be equal to the conclusion.

GBiInt also has the **subformula property**. This is obvious for all rules, except (\rightarrow_R) and the dual $(\neg\leftarrow_L)$. For these, the right premise “constructs” the formulae $\bigwedge \mathcal{P}$ and $\bigvee \mathcal{S}$. However, since \mathcal{P} and \mathcal{S} are sets of sets of subformulae of the conclusion that are again extracted by (\bigwedge_R) and (\bigvee_L) , the right premise of (\rightarrow_R) and $(\neg\leftarrow_L)$ effectively only contains subformulae of the conclusion.

Definition 15 (GBiInt tree). *A GBiInt tree for a sequent*

$$\frac{\mathcal{S}}{\mathcal{P}} \parallel \Gamma \vdash \Delta$$

is a tree rooted at $\gamma = \frac{\mathcal{S}}{\mathcal{P}} \parallel \Gamma \vdash \Delta$, such that:

1. Each child is obtained by a backwards application of a **GBiInt** rule, and
2. Each leaf is an instance of a (\perp_L) , (\top_R) , (Id) or (Ret) rule.

Definition 16. A **GBiInt** tree \mathcal{T} rooted at $\gamma = \frac{\mathcal{P}}{\mathcal{S}} \parallel \Gamma \vdash \Delta$ is a derivation if:

1. γ is the conclusion of a (\perp_L) , (\top_R) or (Id) rule application, OR,
2. γ is the conclusion of a **universal branching** rule application, and **all its premises** are derivations, OR,
3. γ is the conclusion of an **existential branching** rule application, and **some premise** is a derivation.

We say that γ is derivable if there exists a derivation for γ .

We say that γ is not derivable if γ has no derivation.

3.3 Examples

In the following examples, we use a simplified version of the (\wedge_R) rule, which discards the principal formula from the premises, merely to save horizontal space. Also, we only show non-empty variable values.

Example 1. The following is a derivation tree of Uustalu's counterexample, the interaction formula $p \rightarrow (q \vee (r \rightarrow ((p \multimap q) \wedge r)))$, simplified to the sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$. We abbreviate $X := r \rightarrow ((p \multimap q) \wedge r)$. The tree should be read bottom-up while ignoring the variables \mathcal{S} and \mathcal{P} . At the leaves, the variables are assigned and transmit information down to parents and across to some siblings. The top left application of (Ret) occurs because an application of the (\multimap_R) rule to the bolded $p \multimap q$ is blocked, since its left premise would not be different from its conclusion.

Notice that the key to finding the contradiction is the bolded $p \multimap q$ formula that is passed from the left-most leaf node back to the right premise (1) of the (\rightarrow_R) rule. Also, the (\wedge_R) rule in (1) is unary in this case, since the returned \mathcal{P} variable contains only one set of formulae.

$$\begin{array}{c}
 \begin{array}{ccc}
 (Ret) \frac{}{\frac{\mathcal{S}:=\{\{p,r,q\}\} \parallel p, r, q \vdash \mathbf{p} \multimap \mathbf{q}}{\mathcal{P}:=\{\{\mathbf{p} \multimap \mathbf{q}\}\} \parallel p, r, q \vdash \mathbf{p} \multimap \mathbf{q}}} & (Id) \frac{}{p, r \vdash p \multimap q, p} & \\
 (\multimap_R) \frac{}{\frac{\mathcal{S}:=\{\{p,r,q\}\} \parallel p, r \vdash p \multimap q}{\mathcal{P}:=\{\{\mathbf{p} \multimap \mathbf{q}\}\} \parallel p, r \vdash p \multimap q}} & (Id) \frac{}{p, r \vdash r} & \\
 (\wedge_R) \frac{}{\frac{\mathcal{S}:=\{\{p,r,q\}\} \parallel p, r \vdash (p \multimap q) \wedge r}{\mathcal{P}:=\{\{\mathbf{p} \multimap \mathbf{q}\}\} \parallel p, r \vdash (p \multimap q) \wedge r}} & & (1)
 \end{array} \\
 (\rightarrow_R) \frac{}{p \vdash q, r \rightarrow ((p \multimap q) \wedge r)}
 \end{array}$$

Where (1) is:

$$\begin{array}{c}
 (Id) \frac{}{p, \mathbf{q} \vdash q, X, p \multimap q} \quad (Id) \frac{}{p \vdash q, X, p \multimap q, \mathbf{p}} \\
 (\multimap_R) \frac{}{\frac{}{p \vdash q, X, \mathbf{p} \multimap \mathbf{q}}} \\
 (\wedge_R) \frac{}{p \vdash q, X, \wedge\{\{\mathbf{p} \multimap \mathbf{q}\}\}}
 \end{array}$$

Example 2. The following example is a **GBiInt**-tree of a falsifiable sequent, and it shows how in the case of multiple choices for the variables, a contradiction caused by one of them does not give us a derivation. We abbreviate $Y := (\top \prec p) \wedge (\top \prec q)$, and $X := Y \rightarrow \perp$.

$$\begin{array}{c}
\begin{array}{c}
(Ret) \frac{}{S := \{\{X\}\} \parallel X \vdash \perp, \top \prec p} \quad (Ret) \frac{}{S := \{\{X\}\} \parallel X \vdash \perp, \top \prec q} \\
(\wedge_R) \frac{}{\mathcal{P} := \{\{\top \prec p\}\} \parallel \mathcal{P} := \{\{\top \prec q\}\} \parallel X \vdash \perp, Y}
\end{array} \\
(\rightarrow_L) \frac{}{S := \{\{X\}\} \parallel \mathcal{P} := \left\{ \begin{array}{l} \{\top \prec p\}, \\ \{\top \prec q\} \end{array} \right\} \parallel X \vdash \perp, Y} \quad (\perp_L) \frac{}{X, \perp \vdash \perp} \\
(\rightarrow_R) \frac{}{S := \{\{X\}\} \parallel \mathcal{P} := \left\{ \begin{array}{l} \{\top \prec p\}, \\ \{\top \prec q\} \end{array} \right\} \parallel X \vdash \perp} \quad (2) \\
(\rightarrow_R) \frac{}{S := \{\{q\}\} \parallel \mathcal{P} := \left\{ \begin{array}{l} \{p, X \rightarrow \perp\}, \\ \{\top \prec q\} \end{array} \right\} \parallel \vdash p, X \rightarrow \perp}
\end{array}$$

Where (2) is:

$$\begin{array}{c}
(Id) \frac{}{\vdots} \quad (\top_R) \frac{}{\vdots} \quad (Ret) \frac{}{S := \{\{q\}\} \parallel \mathcal{P} := \left\{ \begin{array}{l} \{p, X \rightarrow \perp\}, \\ \{\top \prec q\} \end{array} \right\} \parallel q \vdash p, X \rightarrow \perp, \top \prec q} \quad (\top_R) \frac{}{\vdots} \\
(\neg_R) \frac{}{\vdash p, X \rightarrow \perp, \top \prec p} \quad (\neg_R) \frac{}{S := \{\{q\}\} \parallel \mathcal{P} := \left\{ \begin{array}{l} \{p, X \rightarrow \perp\}, \\ \{\top \prec q\} \end{array} \right\} \parallel \vdash p, X \rightarrow \perp, \top \prec q} \\
(\wedge_R) \frac{}{\mathcal{P} := \left\{ \begin{array}{l} \{p, X \rightarrow \perp\}, \\ \{\top \prec q\} \end{array} \right\} \parallel \vdash p, X \rightarrow \perp, \wedge \left\{ \{\top \prec p\}, \{\top \prec q\} \right\}}
\end{array}$$

In this case, the (\wedge_R) rule in (2) has two premises, since the returned \mathcal{P} variable contains two sets of formulae. Since only the left premise of the (\wedge_R) rule is derivable, the conclusion is not derivable. Thus, the open branch corresponding to the bolded member $\{\top \prec q\}$ of \mathcal{P} remains open. If we did not return both variable choices from the left sibling of (2), then we might mistakenly derive (2) without seeing this open branch.

Lemma 4. *If a **GBiInt**-tree \mathcal{T} rooted at $\gamma = \frac{\mathcal{P}}{\mathcal{S}} \parallel \Gamma \vdash \Delta$ is a derivation then $\mathcal{S} = \mathcal{P} = \epsilon$.*

Proof. By induction on the longest branch in \mathcal{T} .

3.4 Termination Proof

We first show that proof search in **GBiInt** terminates because the subsequent soundness proof relies on our ability to receive the variables from the left premises of transitional rules.

Definition 17. *The rules of **GBiInt** are categorised as follows:*

Operational: (Ret) ;

Function ProveInput: sequent γ_0 Output: Derivable (*true* or *false*)

1. If $\rho \in \{(Id), (\perp_L), (\top_R)\}$ applicable to γ_0 then
 - (a) Return *true*
2. Else if any special or static rule ρ applicable to γ_0 then
 - (a) Let $\gamma_1, \dots, \gamma_n$ be the premises of ρ
 - (b) Return $\bigwedge Prove(\gamma_i)$
3. Else for each transitional rule ρ applicable to γ_0 do
 - (a) Let γ_1 and γ_2 be the premises of ρ
 - (b) If $\bigvee Prove(\gamma_i) = \text{true}$ then return *true*
4. Endif
5. Return *false*.

Fig. 3. Proof search strategy. Note that we have left out the variables for simplicity. $\bigwedge_{i=1}^n Prove(\gamma_i)$ is true iff $Prove(\gamma_i)$ is true for all premises γ_i for $1 \leq i \leq n$, and $\bigvee_{i \in \{1,2\}} Prove(\gamma_i)$ is true iff $Prove(\gamma_i)$ is true for some premise γ_i for $i \in \{1, 2\}$.

Logical:

Static: $(Id), (\perp_L), (\top_R), (\wedge_L), (\vee_L), (\wedge_R), (\vee_R), (\rightarrow_L), (\prec_R), (\rightarrow_R^I), (\prec_L^I)$;

Transitional: $(\rightarrow_R), (\prec_L)$;

Special: $(\vee_L), (\wedge_R)$.

The intuition behind the classification of the logical rules is that the static rules add formulae to the current world in the counter-model, the transitional rules create new worlds and add formulae to them, and the special rules decompose variables returned from non-derivable leaves. We shall prove this formally for each rule later. The classification justifies the following search strategy.

Definition 18 (Strategy). *The strategy defined in Figure 3 is used when applying the rules of our sequent calculus in backward proof search. Note that we have left out the variables for simplicity.*

Definition 19 (Subformulae). *For a BiInt formula, we define the subformulae as follows, where $p \in \text{Atoms}$ and $\varphi, \psi \in \text{Fml}$:*

$$\begin{aligned}
 sf(p) &= \{p\} \\
 sf(\varphi \vee \psi) &= sf(\varphi) \cup sf(\psi) \cup \{\varphi \vee \psi\} \\
 sf(\varphi \wedge \psi) &= sf(\varphi) \cup sf(\psi) \cup \{\varphi \wedge \psi\} \\
 sf(\varphi \rightarrow \psi) &= sf(\varphi) \cup sf(\psi) \cup \{\varphi \rightarrow \psi\} \\
 sf(\varphi \prec \psi) &= sf(\varphi) \cup sf(\psi) \cup \{\varphi \prec \psi\} \\
 sf(\bigvee \mathcal{S}) &= \bigcup_{\Sigma \in \mathcal{S}} sf(\Sigma) \\
 sf(\bigwedge \mathcal{P}) &= \bigcup_{\Pi \in \mathcal{P}} sf(\Pi)
 \end{aligned}$$

For a set Γ of extended **BiInt** formulae, we define $sf(\Gamma) = \bigcup_{\chi \in \Gamma} sf(\chi)$.

Note that the subformulae of $\bigvee \mathcal{S}$ and $\bigwedge \mathcal{P}$ do not include the conjunctions and disjunctions implicit in their **BiInt** equivalents.

Definition 20 (LEN). Let $>_{len}$ be a lexicographic ordering of sequents:

$$(\Gamma_2 \vdash \Delta_2) >_{len} (\Gamma_1 \vdash \Delta_1) \quad \text{iff} \quad \begin{array}{l} |\Gamma_2| > |\Gamma_1| \text{ or} \\ |\Gamma_2| = |\Gamma_1| \text{ and } |\Delta_2| > |\Delta_1| \end{array}$$

Definition 21. Given a **GBiInt**-tree \mathcal{T} and a branch \mathcal{B} in \mathcal{T} , we say that \mathcal{B} is **forward-only** if \mathcal{B} contains only applications of static and special rules, (\rightarrow_R) and the right premises of (\prec_L) . Similarly, \mathcal{B} is **backward-only** if \mathcal{B} contains only applications of static and special rules, (\prec_L) and the right premises of (\rightarrow_R) . A branch is **single-directional** if it is either forward-only or backward-only. Finally, a branch contains **interleaved** left premises of transitional rules if it contains a sequence $\langle \dots, \gamma_i, \dots, \gamma_j, \dots, \gamma_k, \dots \rangle$ such that γ_i is the left premise of (\rightarrow_R) , γ_j is the left premise of (\prec_L) , and γ_k is the left premise of (\rightarrow_R) .

Lemma 5. Every forward-only branch of any **GBiInt**-tree is finite.

Proof. We show that on every such branch, the length of a sequent defined according to $>_{len}$ increases.

Consider a rule ρ , and a backwards application of ρ to some $\Gamma \vdash \Delta$, which yields n premises $\Gamma_i \vdash \Delta_i$, where $1 \leq i \leq n$.

We show that if ρ is a static rule, then for all premises i , we have $(\Gamma_i \vdash \Delta_i) >_{len} (\Gamma \vdash \Delta)$:

$$\begin{array}{l} \rho \in \{(\wedge_L), (\vee_L), (\prec_L^I)\}: \text{ Then } |\Gamma_i| > |\Gamma|; \\ \rho = (\rightarrow_R^I): \text{ Then } |\Gamma_1| = |\Gamma| \text{ and } |\Delta_1| > |\Delta|; \\ \rho = (\rightarrow_L): \text{ Then for the left premise, } |\Gamma_1| = |\Gamma| \text{ and } |\Delta_1| > |\Delta|, \text{ and for the} \\ \quad \text{right premise, } |\Gamma_2| > |\Gamma|; \\ \rho = (\prec_R): \text{ Then for the left premise, } |\Gamma_1| > |\Gamma|, \text{ and for the right premise,} \\ \quad |\Gamma_2| = |\Gamma| \text{ and } |\Delta_2| > |\Delta|. \end{array}$$

We now show the cases for $\rho \in \{(\rightarrow_R), (\prec_L), (\wedge_R), (\vee_L)\}$. Even though the right premise of (\rightarrow_R) and (\prec_L) itself is not greater than the conclusion, we show that the lemma holds on the overall **GBiInt** branch, since according to the strategy we immediately apply (\wedge_R) or (\vee_L) , thus increasing the length of the premise according to $>_{len}$.

$\rho = (\rightarrow_R)$: For every (\rightarrow_R) rule application:

1. Consider the left premise $\Gamma_1 \vdash \Delta_1$. We know that according to our strategy, the (\rightarrow_R^I) rule has already been applied and thus $\psi \in \Delta$, so (\rightarrow_R) is applied only if $\varphi \notin \Gamma$. Therefore, for the left premise, we have $|\Gamma_1| > |\Gamma|$;

2. Consider the right premise $\Gamma_2 \vdash \Delta_2$. It is created only if

$$\mathcal{P}_1 \neq \epsilon \ \& \ \forall \Pi_i \in \mathcal{P}_1. \Pi_i \not\subseteq \{\Delta, \varphi \rightarrow \psi\}. \quad (3.1)$$

That is, every member of \mathcal{P}_1 introduces new formulae to the RHS. But recall that $sf(\bigwedge \mathcal{P}_1) \subseteq sf(\Gamma \cup \Delta)$. According to our strategy, the (\bigwedge_R) rule will be immediately applied to $\bigwedge \mathcal{P}_1$ in Δ_2 , giving $n \geq 1$ premises $\Gamma_2^j \vdash \Delta_2^j$ where $1 \leq j \leq n$. By 3.1, we will then have $|\Delta_2^j| > |\Delta|$ for all j . We also have $|\Gamma_2^j| = |\Gamma|$ for all j . Therefore, according to the lexicographic ordering, we have $(\Gamma_2^j \vdash \Delta_2^j) >_{len} (\Gamma \vdash \Delta)$ for all the premises $\Gamma_2^j \vdash \Delta_2^j$.

$\rho = (\bigwedge_R)$: Since the (\bigwedge_R) rule is only used in conjunction with the right premise of the (\rightarrow_R) rule, see case 2 above;

$\rho = (\neg_L)$: For every (\neg_L) rule application:

1. The assumption of the lemma does not apply to the left premise;
2. The case for the right premise is dual to the case for (\rightarrow_R) above.

$\rho = (\bigvee_L)$: By symmetry with the case for (\bigwedge_R) above;

Since the length of a sequent defined according to $>_{len}$ increases on every forward-only branch as shown above, and since **GBiInt** has the subformula property, eventually no more formulae can be added to a sequent on a forward-only branch, and the branch will terminate.

Lemma 6. *Every backward-only branch of any **GBiInt**-tree is finite.*

Proof. By symmetry with Lemma 5.

Lemma 7. *If a **GBiInt**-tree contains an infinite branch, then the branch contains an infinite number of interleaved left premises of transitional rules.*

Proof. By Lemmas 5 and 6, single-directional branches must eventually terminate. Thus, a potential infinite loop must involve an infinite number of interleaved left premises of transitional rules (\rightarrow_R) and (\neg_L) .

Definition 22 (Degree). *The degree of a BiInt formula χ is defined as:*

$$\deg(\chi) = \begin{cases} 0 & \text{if } \chi \in \text{Atoms} \\ \deg(\varphi) + \deg(\psi) & \text{if } \chi \in \{\varphi \vee \psi, \varphi \wedge \psi\} \\ \deg(\varphi) + \deg(\psi) + 1 & \text{if } \chi \in \{\varphi \rightarrow \psi, \varphi \neg \psi\} \end{cases}$$

Thus, the degree of φ is the number of \rightarrow and \neg connectives in φ .

The degree of a sequent $\Gamma \vdash \Delta$ is defined as:

$$\deg(\Gamma \vdash \Delta) = \sum_{\varphi \in sf(\Gamma \cup \Delta)} \deg(\varphi)$$

Note that we have deliberately defined the degree of a sequent as the sum of the degrees of subformulae, because it allows us to make the following observations, which will be crucial in the main termination proof.

Corollary 1. *Since **GBiInt** has the subformula property, the degree of a sequent can never increase in backward proof search. In other words, no **GBiInt** rule can increase the degree of a sequent.*

Corollary 2. *Given two sequents γ_1 and γ_2 , if $sf(\gamma_2) \subsetneq sf(\gamma_1)$, then $deg(\gamma_2) < deg(\gamma_1)$. That is, removing some formula φ from a sequent during backward proof search decreases the degree of the sequent if φ is not a subformula of any other formula in the sequent, since φ no longer contributes to the sum of degrees of subformulae.*

Theorem 1 (Termination). *Every **GBiInt**-tree constructed according to the strategy of Definition 18 is finite.*

Proof. Suppose for a contradiction that there exists an infinite **GBiInt**-tree \mathcal{T} . Since every rule has a finite number of premises, i.e., finite branching, then by König's lemma an infinite tree can only be obtained by having a branch of infinite length. Thus, \mathcal{T} has an infinite branch \mathcal{B} . By Lemma 7, \mathcal{B} must contain an infinite number of interleaved left premises of transitional rules, as shown below:

$$\begin{array}{c}
 \vdots \\
 \hline
 (\rightarrow_R) \frac{\frac{\pi_2 = (I_2, \varphi_2 \vdash \psi_2)}{I_2 \vdash \Delta_2, \varphi_2 \rightarrow \psi_2} \quad \frac{\vdots}{\pi_2^s}}{\vdots} \\
 \\
 \vdots \\
 \hline
 (\prec_L) \frac{\frac{\varphi_1 \vdash \psi_1, \Delta_1}{I_1, \varphi_1 \prec \psi_1 \vdash \Delta_1} \quad \frac{\vdots}{\pi_1^r}}{\vdots} \\
 \\
 \vdots \\
 \hline
 (\rightarrow_R) \frac{\frac{I_0, \varphi_0 \vdash \psi_0}{\pi_0 = (I_0 \vdash \Delta_0, \varphi_0 \rightarrow \psi_0)} \quad \frac{\vdots}{\pi_0^r}}{\vdots} \\
 \vdots
 \end{array}$$

Let $\chi \in sf(\pi_0)$ be some formula such that $deg(\chi) = \max(\{deg(\varphi) \mid \varphi \in sf(\pi_0)\})$, that is, χ is one of the subformulae with the maximum degree. In particular, this means that χ is not a subformula of any formula with a larger degree. We shall now show that $\chi \notin sf(\pi_2)$.

There are two cases:

- $\chi \notin sf(I_0)$: Then $\chi \in sf(\Delta_0)$ or $\chi = \varphi_0 \rightarrow \psi_0$. In both cases, $\chi \notin sf(\pi_2)$.
- $\chi \in sf(I_0)$: Then it cannot be the case that $\chi \in sf(\varphi_1)$ or $\chi \in sf(\psi_1)$, since then $deg(\varphi_1 \prec \psi_1) > deg(\chi)$, contradicting our assumption that $deg(\chi) = \max(\{deg(\varphi) \mid \varphi \in sf(\pi_0)\})$. Therefore, either:
 - χ and all its occurrences in subformulae disappear from the sequent at the premise of (\prec_L) , in which case $\chi \notin sf(\pi_2)$, or

- χ is moved to the RHS of the sequent by applying the (\rightarrow_L) rule to some formula $\chi \rightarrow \tau$. However, since $\deg(\chi \rightarrow \tau) > \deg(\chi)$, it again contradicts our assumption that $\deg(\chi) = \max(\{\deg(\varphi) \mid \varphi \in sf(\pi_0)\})$.

We have shown that for some formula χ we have $\chi \in sf(\pi_0)$ and $\chi \notin sf(\pi_2)$. Also, by the subformula property of **GBiInt** we have $sf(\pi_2) \subseteq sf(\pi_0)$. Together with $\chi \in sf(\pi_0)$ and $\chi \notin sf(\pi_2)$, this means $sf(\pi_2) \subsetneq sf(\pi_0)$. Then by Corollary 2 we have $\deg(\pi_2) < \deg(\pi_0)$. Note that the steps indicated by vertical ellipses (\vdots) are arbitrary, since by Corollary 1 no rule can increase the degree of a sequent.

Since we have $\deg(\pi_2) < \deg(\pi_0)$, we know that every sequence of interleaved transitional rule applications must decrease the degree of the sequent. This can only happen a finite number of times, until no more transitional rules are applicable. Therefore our assumption was wrong, and no branch \mathcal{B} can be infinite. Therefore, every **GBiInt**-tree is finite.

4 Soundness

4.1 Proof Outline

Instead of the traditional approach of showing that each rule application preserves validity downwards, we use the notion of falsifiability and show that each rule application preserves falsifiability upwards. We then use Lemma 3 to make the connection between falsifiability and validity.

Also, our addition of variables to the calculus introduces a two-way flow of information in the **GBiInt** trees, and this complicates the usually simple soundness proof.

We separate the notion of soundness into two: *local soundness*, applicable locally to a single rule application, and *global soundness*, which takes into account the propagation of variables from the leaves down to some node, and possible instances of the operational (*Ret*) rule. Note that locality here refers to locality in the **GBiInt** trees, not locality in the underlying Kripke models. We use the notions of static and transitional rules to classify the rules according to this latter notion.

4.2 Local soundness

Definition 23 (Local soundness). *A logical rule in **GBiInt** is locally sound if and only if:*

- For rules with **universal** branching: if the conclusion is falsifiable, then some premise is falsifiable;
- For rules with **existential** branching: if the conclusion is falsifiable, then all premises are falsifiable.

We shall now show that each static and special rule is locally sound, and we shall then use induction on the height of a derivation tree to extend our proof to arbitrary trees containing static rules, special rules, transitional rules and the operational (*Ret*) rule.

Lemma 8. *Each static and special rule of **GBiInt** is locally sound.*

Proof. We consider each static and special rule in turn. We assume that the conclusion is falsifiable, and show that some premise is falsifiable.

$$1. (Id) \frac{}{\mathcal{S}:=\epsilon \parallel \Gamma, \varphi \vdash \Delta, \varphi \atop \mathcal{P}:=\epsilon}$$

The conclusion of this rule is never falsifiable, because no **BiInt** model can contain a world w such that $w \models \varphi$ and $w \not\models \varphi$.

$$2. (\perp_L) \frac{}{\mathcal{S}:=\epsilon \parallel \Gamma, \perp \vdash \Delta \atop \mathcal{P}:=\epsilon}$$

The conclusion of this rule is never falsifiable, because by Property 5 of Definition 4, no **BiInt** model can contain a world w such that $w \models \perp$.

$$3. (\top_R) \frac{}{\mathcal{S}:=\epsilon \parallel \Gamma \vdash \Delta, \top \atop \mathcal{P}:=\epsilon}$$

The conclusion of this rule is never falsifiable, because by Property 4 of Definition 4, no **BiInt** model can contain a world w such that $w \not\models \top$.

$$4. (\wedge_R) \frac{\mathcal{S}_1 \parallel \Gamma \vdash \Delta, \varphi \wedge \psi, \varphi \quad \mathcal{S}_2 \parallel \Gamma \vdash \Delta, \varphi \wedge \psi, \psi}{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2 \parallel \Gamma \vdash \Delta, \varphi \wedge \psi \atop \mathcal{P}:=\mathcal{P}_1 \cup \mathcal{P}_2}$$

Since the conclusion is falsifiable by assumption, we know from Definition 12 that there exists a world w_0 such that:

- (i) $w_0 \models \Gamma$ and
- (ii) $w_0 \models \Delta, \varphi \wedge \psi$.

From the semantics of \wedge in **BiInt**, (b) implies that either:

- (ii.1) $w_0 \models \Delta, \varphi \wedge \psi, \varphi$ or
- (ii.2) $w_0 \models \Delta, \varphi \wedge \psi, \psi$.

To show that some premise of the (\wedge_R) rule is falsifiable, we need to show that there exists a world w' such that some premise is falsifiable at w' . We let $w' = w_0$.

Then case (ii.1) together with (i) gives us that the left premise is falsifiable, or case (ii.2) together with (i) gives us that the right premise is falsifiable.

$$5. (\vee_L) \frac{\mathcal{S}_1 \parallel \Gamma, \varphi \vee \psi, \varphi \vdash \Delta \quad \mathcal{S}_2 \parallel \Gamma, \varphi \vee \psi, \psi \vdash \Delta}{\mathcal{S}:=\mathcal{S}_1 \cup \mathcal{S}_2 \parallel \Gamma, \varphi \vee \psi \vdash \Delta \atop \mathcal{P}:=\mathcal{P}_1 \cup \mathcal{P}_2}$$

By symmetry with the (\wedge_R) rule.

6.

$$(\vee_R) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma \vdash \Delta, \varphi \vee \psi, \varphi, \psi}{\frac{\mathcal{S} := \mathcal{S}_1}{\mathcal{P} := \mathcal{P}_1} \parallel \Gamma \vdash \Delta, \varphi \vee \psi}$$

Since the conclusion is falsifiable by assumption, we know from Definition 12 that there exists a world w_0 such that:

- (i) $w_0 \models \Gamma$ and
- (ii) $w_0 \models \Delta, \varphi \vee \psi$

To show that the premise of the (\vee_R) rule is falsifiable, we need to show that there exists a world w' such that the premise is falsifiable at w' . We let $w' = w_0$.

From the semantics of \vee in **BiInt**, (ii) implies that $w_0 \models \Delta, \varphi \vee \psi, \varphi$ and $w_0 \models \Delta, \varphi \vee \psi, \psi$. Together with (i), this means that the premise is falsifiable.

7.

$$(\wedge_L) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma, \varphi \wedge \psi, \varphi, \psi \vdash \Delta}{\frac{\mathcal{S} := \mathcal{S}_1}{\mathcal{P} := \mathcal{P}_1} \parallel \Gamma, \varphi \wedge \psi \vdash \Delta}$$

By symmetry with the (\vee_R) rule.

8.

$$(\rightarrow_L) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma, \varphi \rightarrow \psi \vdash \varphi, \Delta \quad \frac{\mathcal{S}_2}{\mathcal{P}_2} \parallel \Gamma, \varphi \rightarrow \psi, \psi \vdash \Delta}{\frac{\mathcal{S} := \mathcal{S}_1 \cup \mathcal{S}_2}{\mathcal{P} := \mathcal{P}_1 \cup \mathcal{P}_2} \parallel \Gamma, \varphi \rightarrow \psi \vdash \Delta}$$

Since the conclusion is falsifiable by assumption, we know from Definition 12 that there exists a world w_0 such that:

- (i) $w_0 \models \Gamma, \varphi \rightarrow \psi$ and
- (ii) $w_0 \models \Delta$.

From the semantics of \rightarrow in **BiInt**, (i) implies that for all successors w , we have $w \not\models \varphi$ or $w \models \psi$.

By reflexivity of \mathcal{R} , this applies to w_0 too, so we have:

- (i.1) $w_0 \not\models \varphi$ or
- (i.2) $w_0 \models \psi$.

To show that some premise of the (\rightarrow_L) rule is falsifiable, we need to show that there exists a world w' such that some premise is falsifiable at w' . We let $w' = w_0$.

Then items (i), (ii) and (i.1) give us that the left premise is falsifiable, or items (i), (ii) and (i.2) give us that the right premise is falsifiable.

9.

$$(\prec_R) \frac{\frac{\mathcal{S}_1}{\mathcal{P}_1} \parallel \Gamma, \psi \vdash \Delta, \varphi \prec \psi \quad \frac{\mathcal{S}_2}{\mathcal{P}_2} \parallel \Gamma \vdash \Delta, \varphi \prec \psi, \varphi}{\frac{\mathcal{S} := \mathcal{S}_1 \cup \mathcal{S}_2}{\mathcal{P} := \mathcal{P}_1 \cup \mathcal{P}_2} \parallel \Gamma \vdash \Delta, \varphi \prec \psi}$$

By symmetry with (\rightarrow_L) .

10.

$$(\rightarrow_R^I) \frac{\frac{\mathcal{S}_1 \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi, \psi}{\mathcal{S} := \mathcal{S}_1 \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi}}{\mathcal{P} := \mathcal{P}_1 \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi}$$

Since the conclusion is falsifiable by assumption, we know from Definition 12 that there exists a world w_0 such that:

- (i) $w_0 \models \Gamma$ and
- (ii) $w_0 \models \Delta, \varphi \rightarrow \psi$.

From the semantics of \rightarrow in **BiInt**, (ii) implies that there exists a successor w_1 such that:

- (iii) $w_0 \mathcal{R} w_1$ and
- (iv) $w_1 \models \varphi$ and
- (v) $w_1 \not\models \psi$.

Then, by the reverse persistence property of **BiInt**, and (iii) and (v), we have:

- (vi) $w_0 \not\models \psi$.

To show that the premise of the (\rightarrow_R) rule is falsifiable, we need to show that there exists a world w' such that the premise is falsifiable at w' . We let $w' = w_0$.

Then items (i), (ii) and (vi) give us that the premise is falsifiable.

11.

$$(\prec_L^I) \frac{\frac{\mathcal{S}_1 \parallel \Gamma, \varphi, \varphi \prec \psi \vdash \Delta}{\mathcal{S} := \mathcal{S}_1 \parallel \Gamma, \varphi \prec \psi \vdash \Delta}}{\mathcal{P} := \mathcal{P}_1 \parallel \Gamma, \varphi \prec \psi \vdash \Delta}$$

By symmetry with (\rightarrow_R^I) .

12.

$$(\bigvee_L) \frac{\frac{\mathcal{S}_1 \parallel \Gamma, \Sigma_1 \vdash \Delta \quad \dots \quad \mathcal{S}_n \parallel \Gamma, \Sigma_n \vdash \Delta}{\mathcal{S} := \bigcup_{i=1}^n \mathcal{S}_i \parallel \Gamma, \bigvee \Sigma \vdash \Delta}}{\mathcal{P} := \bigcup_{i=1}^n \mathcal{P}_i \parallel \Gamma, \bigvee \Sigma \vdash \Delta}$$

Since the conclusion is falsifiable by assumption, we know from Definition 12 that there exists a world w_0 such that:

- (i) $w_0 \models \Gamma, \bigvee \Sigma$ and
- (ii) $w_0 \models \Delta$.

From the semantics of $\bigvee \Sigma$ (recall Definition 10), (i) implies that:

- (iii) for some $\Sigma_i \in \Sigma$, we have $w_0 \models \Sigma_i$.

To show that some premise of the (\bigvee_L) rule is falsifiable, we need to show that there exists a world w' such that this premise is falsifiable at w' . We let $w' = w_0$.

Then items (i), (ii) and (iii) give us that the i -th premise containing Σ_i is falsifiable at w_0 .

13.

$$(\bigwedge_R) \frac{\frac{\mathcal{S}_1 \parallel \Gamma \vdash \Delta, \Pi_1 \quad \dots \quad \mathcal{S}_n \parallel \Gamma \vdash \Delta, \Pi_n}{\mathcal{S} := \bigcup_{i=1}^n \mathcal{S}_i \parallel \Gamma \vdash \Delta, \bigwedge \Pi}}{\mathcal{P} := \bigcup_{i=1}^n \mathcal{P}_i \parallel \Gamma \vdash \Delta, \bigwedge \Pi}$$

By symmetry with (\bigvee_L) .

Remark 1. Note that the static rules also preserve falsifiability downwards: if some premise π is falsifiable, then the conclusion γ is falsifiable. This is easy to see, since we have $LHS_\pi \supseteq LHS_\gamma$ and $RHS_\pi \supseteq RHS_\gamma$.

4.3 Global soundness

We have shown that all the static and special rules preserve falsifiability upwards, in other words, they are *locally sound*. Since the \mathcal{S} and \mathcal{P} variables propagate downwards, from the leaves to the root, we can only reason about the variable conditions of rules when we consider an entire tree rooted at a rule application. Similarly, since the soundness of the transitional rules relies on the variables, we can only reason about it we consider an entire tree rooted at a transitional rule application. We shall now show that **GBiInt** rules are *globally sound*, that is, they preserve falsifiability upwards *and* variable conditions downwards.

Lemma 9 (Global soundness). *Given any **GBiInt** tree \mathcal{T} , for every sequent $\gamma_0 \in \mathcal{T}$, the following holds: if γ_0 is falsifiable, then:*

1. *Some universally branching, or all existentially branching, premises are falsifiable,*
2. *The variable conditions hold at γ_0 .*

Proof. By induction on the length $h(\gamma_0)$ of the longest branch from γ_0 to a leaf sequent of \mathcal{T} .

Base case: $h(\gamma_0) = 0$. So γ_0 itself is an instance of (Id) , (\perp_L) , (\top_R) , or (Ret) .

(Id) , (\perp_L) , (\top_R) : The conclusion of these rules is never falsifiable, so there is nothing to show.

(Ret) :

The conclusion of the (Ret) rule is $\Gamma \vdash \Delta$, and there is no premise. From the side condition of the (Ret) rule, we know that no other rules are applicable to $\Gamma \vdash \Delta$. We will now show that $\Gamma \vdash \Delta$ is falsifiable, and that it obeys the variable conditions.

We create a model with a single world w_0 , and for every atom p in Γ , we let $\vartheta(w_0, p) = \text{true}$, and for every atom q in Δ , we let $\vartheta(w_0, q) = \text{false}$. Note that an atom cannot be both in Γ and Δ , since the (Id) rule in particular is not applicable to $\Gamma \vdash \Delta$.

To show that $\Gamma \vdash \Delta$ is falsifiable at w_0 , we need to show that $w_0 \models \Gamma$ and $w_0 \not\models \Delta$. For every atom in Γ and Δ , the valuation ensures this. For every composite formula φ , we do a simple induction on its length. The fact that the (Ret) rule is applied implies that no other rules are applicable, therefore the required subformula ψ is already in Γ or Δ as appropriate, and ψ falls under the induction hypothesis.

Thus we know that:

- (i) $w_0 \models \Gamma$ and

(ii) $w_0 \models \Delta$.

Then (i) and the persistence property of **BiInt** give us that $\forall w \in \mathcal{W} : w_0 \mathcal{R} w \Rightarrow w \models \Gamma$. Similarly, (ii) and the reverse persistence property of **BiInt** give us that $\forall w \in \mathcal{W} . w \mathcal{R} w_0 \Rightarrow w \models \Delta$. Then the conclusion of the (*Ret*) rule obeys the variable conditions:

S-condition: Successor condition

$$\exists \Sigma \in \{\Gamma\}. \forall w \in \mathcal{W} . w_0 \mathcal{R} w \Rightarrow w \models \Sigma$$

P-condition: Predecessor condition

$$\exists \Pi \in \{\Delta\}. \forall w \in \mathcal{W} . w \mathcal{R} w_0 \Rightarrow w \models \Pi$$

Induction step: We assume that the lemma holds for all γ_0 with $h(\gamma_0) \leq k$, and show that it holds for all γ_0 with $h(\gamma_0) \leq k + 1$.

Consider the rule application ρ such that γ_0 is the conclusion of ρ . By the assumption of the lemma, we have that the conclusion γ_0 of ρ is falsifiable at some w_0 in some model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$. The only possibilities are that ρ is a static or a special rule, or that it is a transitional rule:

1. ρ is one of the static or special rules (universally branching). Then Lemma 8 tells us that some premise is falsifiable. We now need to show that the variable conditions hold at γ_0 . There are two cases:

ρ is **unary**: The premise γ_1 of ρ has $h(\gamma_1) \leq k$, therefore the induction hypothesis applies to γ_1 . By Lemma 8 and the fact that γ_0 is falsifiable at w_0 , we know that the premise γ_1 is falsifiable at w_0 , so by the induction hypothesis we have that the variable conditions hold at γ_1 . Since γ_1 has the same variables as γ_0 , and since γ_1 is falsified by the same world w_0 as γ_0 , we then know that γ_0 also obeys the variable conditions.

ρ is **n -ary with $n > 1$** : We show the case for \mathcal{S} ; the case for \mathcal{P} is symmetric. The premises γ_1 to γ_n of ρ each have $\gamma_i \leq k$, therefore the induction hypothesis applies to each γ_i . By Lemma 8 and the fact that γ_0 is falsifiable at w_0 , we know that some γ_m is falsifiable at w_0 , too. Therefore the induction hypothesis tells us that the variable conditions hold at γ_m . That is, we know that:

$$\exists \Sigma_m \in \mathcal{S}_m. \forall w \in \mathcal{W} . w_0 \mathcal{R} w \Rightarrow w \models \Sigma_m.$$

To show that the conclusion γ_0 obeys the variable condition for \mathcal{S} , we need to show the following:

$$\exists \Sigma \in \bigcup_1^n \mathcal{S}_i. \forall w \in \mathcal{W} . w_0 \mathcal{R} w \Rightarrow w \models \Sigma.$$

Since $\Sigma_m \in \mathcal{S}_m$ and $\mathcal{S}_m \subseteq \bigcup_1^n \mathcal{S}_i$, we have $\Sigma_m \in \bigcup_1^n \mathcal{S}_i$ and thus the variable conditions hold for \mathcal{S} at the conclusion γ_0 .

2. ρ is one of the transitional rules (existentially branching). We show the case for the (\rightarrow_R) rule, the case for the (\prec_L) rule is symmetric:

$$\begin{array}{c}
 \begin{array}{c} \mathcal{S}_1 \\ \mathcal{P}_1 \end{array} \parallel \Gamma, \varphi \vdash \psi \quad \begin{array}{c} \mathcal{S}_2 \\ \mathcal{P}_2 \end{array} \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi, \wedge \mathcal{P}_1 \\
 \hline
 (\rightarrow_R) \quad \text{---} \quad \left\{ \begin{array}{ll} \mathcal{S}_1 / \mathcal{P}_1 & \text{if } \mathcal{P}_1 = \epsilon \\ \mathcal{S}_2 / \mathcal{P}_2 & \text{if right prem created} \\ \{\Gamma\} / \{\Delta, \varphi \rightarrow \psi\} & \text{otherwise} \end{array} \right\} \parallel \Gamma \vdash \Delta, \varphi \rightarrow \psi \\
 \text{right prem created only if } \mathcal{P}_1 \neq \epsilon \ \& \ \forall \Pi_i \in \mathcal{P}_1 . \Pi_i \not\subseteq \{\Delta, \varphi \rightarrow \psi\}
 \end{array}$$

So suppose that the conclusion is falsifiable. Then we know from Definition 12 that there exists a world w_0 such that:

- (i) $w_0 \models \Gamma$ and
- (ii) $w_0 \models \Delta, \varphi \rightarrow \psi$.

From the semantics of \rightarrow in **BiInt**, (ii) implies that there exists a successor w_1 such that:

- (iii) $w_0 \mathcal{R} w_1$ and
- (iv) $w_1 \models \varphi$ and
- (v) $w_1 \not\models \psi$.

- (a) To show that the left premise of the (\rightarrow_R) rule is falsifiable, we need to show that there exists a world w' such that this premise is falsifiable at w' . We let $w' = w_1$.

Then items (i), (iv) and (v) give us that the left premise is falsifiable. Now, the left premise γ_1 is of distance $\leq k$ from the furthest leaf node of \mathcal{T} , therefore the induction hypothesis applies to γ_1 . By the hypothesis assumption, since γ_1 is falsifiable at w_1 , we have that the variable conditions hold at γ_1 . In particular, the \mathcal{P} condition holds, giving us:

$$\exists \Pi \in \mathcal{P}_1. \forall w \in \mathcal{W}. w \mathcal{R} w_1 \Rightarrow w \models \Pi \quad (4.1)$$

Now there are two cases: either the right premise was created, or it was not (and there is nothing to show). If it was created, then we need to show that it is falsifiable by exhibiting a world w'' such that the right premise is falsifiable at w'' . We let $w'' = w_0$. Then, since $w_0 \mathcal{R} w_1$, we have $w_0 \models \Pi$ by (4.1). Since $\Pi \in \mathcal{P}_1$, then by Definition 10 we have that $w_0 \models \bigwedge \mathcal{P}_1$. Together with (i) and (ii), this means that the right premise is falsifiable at w_0 .

Moreover, the variable conditions hold at the right premise, since it also is falsifiable, and of distance $\leq k$ from the furthest leaf node of \mathcal{T} , so the induction hypothesis applies to it.

- (b) We need to show that the variable conditions hold at the conclusion γ_0 of the (\rightarrow_R) rule. We show the case for the variable \mathcal{S} ; the case for \mathcal{P} is symmetric. We need to show that:

$$\exists \Sigma \in \mathcal{S}. \forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \models \Sigma \quad (4.2)$$

Where $\mathcal{S} := \begin{cases} \mathcal{S}_1 & \text{if } \mathcal{P}_1 = \epsilon \\ \mathcal{S}_2 & \text{if right prem created} \\ \{\Gamma\} & \text{otherwise} \end{cases}$

Since we have shown that the variable conditions hold at the left premise, we know that in particular $\mathcal{P}_1 \neq \epsilon$. Therefore there are two cases: either the right premise was created, or it was not:

- If the right premise γ_2 was created, then we know that the variable conditions hold at γ_2 , since γ_2 falls under the induction hypothesis. This gives us:

$$\exists \Sigma_2 \in \mathcal{S}_2. \forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \models \Sigma_2$$

Thus $\mathcal{S} := \mathcal{S}_2$ obeys (4.2).

- If the right premise was not created, then we need to show that $\{\Gamma\}$ obeys the variable conditions at the conclusion. Now, we have $w_0 \models \Gamma$ by (i), and then the persistence property tells us that $\forall w \in \mathcal{W}. w_0 \mathcal{R} w \Rightarrow w \models \Gamma$. Thus $\mathcal{S} := \{\Gamma\}$ obeys (4.2).

4.4 Main Soundness Proof

Lemma 10. *If $\Gamma \vdash \Delta$ is derivable then $\Gamma \vdash \Delta$ is not falsifiable.*

Proof. By induction on the height k of the derivation.

Base case: For the base case, the height is 1. A derivation of height 1 can only be an instance of (\perp_L) , (\top_R) or (Id) . In each case, γ is not falsifiable, as shown in cases 1 to 3 of Lemma 8.

Inductive step: We assume that if there is a derivation for γ of height $\leq k$, then γ is not falsifiable. We show that if there is a derivation for γ of height $\leq k+1$, then γ is not falsifiable.

For a contradiction, suppose there is a derivation \mathcal{T} for γ of height $k+1$ and γ is falsifiable. Consider the bottom-most rule application ρ in \mathcal{T} , then γ is the conclusion of ρ .

Then, by Definition 16, since \mathcal{T} is a derivation, then all universally branching premises, or some existentially branching premise of ρ are rooted at derivations of height $\leq k$, so by the induction hypothesis, all universally branching premises are, or some existentially branching premise is not falsifiable. But since the conclusion γ of ρ is falsifiable by supposition, then by Lemma 9, some universally branching premise, or all existentially branching premises are falsifiable. Now we have a contradiction, therefore our assumption was wrong and γ is not falsifiable.

Theorem 2 (Soundness). *If $\Gamma \vdash \Delta$ is derivable, then $\Gamma \Vdash_{\text{BiInt}} \Delta$.*

Proof. By Lemma 10, we have that $\Gamma \vdash \Delta$ is not falsifiable. Then by Lemma 3, we have $\Gamma \Vdash_{\text{BiInt}} \Delta$.

5 Completeness

5.1 Proof Outline

We wish to prove:

if $\Gamma \Vdash_{\text{BiInt}} \Delta$, then $\Gamma \vdash \Delta$ is derivable.

Instead, we prove the contrapositive:

if $\Gamma \vdash \Delta$ is not derivable, then there exists a counter-model for $\Gamma \Vdash_{\text{BiInt}} \Delta$.

Our proof is based on a standard technique for proving completeness of tableau calculi: see [9]. We have adapted this technique to a two-sided sequent calculus with variables.

We assume that $\Gamma \vdash \Delta$ is not derivable, meaning that none of the **GBiInt**-trees for $\Gamma \vdash \Delta$ is a derivation. Then we choose formulae from sequents found in possibly different **GBiInt**-trees for $\Gamma \vdash \Delta$ in order to construct a counter-model for $\Gamma \Vdash_{\text{BiInt}} \Delta$. The counter-model is constructed so that it contains a world w_0 such that $w_0 \models \Gamma$ and $w_0 \not\models \Delta$, hence $\Gamma \Vdash_{\text{BiInt}} \Delta$ does not hold.

5.2 Saturated Sets

Definition 24. Given a sequent $\Gamma \vdash \Delta$, we say that:

- $\Gamma \vdash \Delta$ is **consistent** if all of the following hold:
 1. $\perp \notin \Gamma$
 2. $\top \notin \Delta$
 3. $\Gamma \cap \Delta = \epsilon$
- $\Gamma \vdash \Delta$ is **closed** with respect to a **GBiInt** rule ρ if either:
 - ρ is not applicable to $\Gamma \vdash \Delta$, or
 - Whenever $\Gamma \vdash \Delta$ matches the conclusion of an instance of ρ , then for some premise $\Gamma_1 \vdash \Delta_1$ of the instance of ρ , we have $\Gamma_1 \subseteq \Gamma$ and $\Delta_1 \subseteq \Delta$.
- $\Gamma \vdash \Delta$ is **saturated** if it is consistent and closed with respect to the static rules of **GBiInt**.

The following corollaries follow directly from the definition of consistent sequents.

Corollary 3. If $\Gamma \vdash \Delta$ is consistent, then none of the rules (Id) , (\perp_L) , (\top_R) is applicable to it.

Corollary 4. If the sequent

$$\frac{\mathcal{S}}{\mathcal{P}} \parallel \Gamma \vdash \Delta$$

is not derivable, then $\Gamma \vdash \Delta$ is consistent for all values of \mathcal{S} and \mathcal{P} .

Remark 2. As usual, every sequent has a set of one or more “saturations” due to the branching of (\wedge_R) , (\vee_L) , etc., rules. The usual approach is to non-deterministically choose one of the non-derivable premises of each such rule. However, in the presence of the inverse relation, a branch that appears open may close once we return variables to a lower sequent. Therefore, we need to temporarily keep all the non-derivable premises, since we do not know which of the open branches will stay open when we return to a lower sequent.

Lemma 11. For each finite non-derivable sequent $\Gamma \vdash \Delta$, there is an effective procedure to construct a finite set $\zeta = \{\alpha_1, \dots, \alpha_n\}$ of finite saturated sequents, with $\Gamma \cup \Delta \subseteq LHS(\alpha_j) \cup RHS(\alpha_j) \subseteq sf(\Gamma) \cup sf(\Delta)$ for all $1 \leq j \leq n$.

Proof. Since $\Gamma \vdash \Delta$ is non-derivable, we know from Corollary 4 that $\Gamma \vdash \Delta$ is consistent. Then from Corollary 3 we know that the (Id) , (\perp_L) , (\top_R) rules are not applicable to $\Gamma \vdash \Delta$. Let $\mathcal{T} = \Gamma \vdash \Delta$. While some static rule ρ is applicable to a leaf of \mathcal{T} , extend \mathcal{T} by applying ρ to the leaf to obtain new leaves. Keep the non-derivable leaves only; by Corollary 4 they are consistent. By Theorem 1, the saturation process will eventually terminate; let $\zeta = \{\alpha_1, \dots, \alpha_n\}$ be the final leaves of \mathcal{T} . Since the formulae in each premise are always subformulae of the conclusion, we have that $LHS(\alpha_j) \cup RHS(\alpha_j) \subseteq sf(\Gamma) \cup sf(\Delta)$ for all $1 \leq j \leq n$.

5.3 Model Graphs and Satisfiability Lemma

We shall use model graphs as an intermediate structure between **GBiInt**-trees and **BiInt** models.

Definition 25. A model graph for a sequent $\Gamma \vdash \Delta$ is a finite **BiInt** frame $\langle \mathcal{W}, \mathcal{R} \rangle$ such that all $w \in \mathcal{W}$ are saturated sequents $\Gamma_w \vdash \Delta_w$ and all of the following hold:

1. $\Gamma \subseteq \Gamma_{w_0}$ and $\Delta \subseteq \Delta_{w_0}$ for some $w_0 \in \mathcal{W}$, where $w_0 = \Gamma_{w_0} \vdash \Delta_{w_0}$;
2. if $\varphi \rightarrow \psi \in \Delta_w$ then $\exists v \in \mathcal{W}$ with $w\mathcal{R}v$ and $\varphi \in \Gamma_v$ and $\psi \in \Delta_v$;
3. if $\varphi \prec \psi \in \Gamma_w$ then $\exists v \in \mathcal{W}$ with $v\mathcal{R}w$ and $\varphi \in \Gamma_v$ and $\psi \in \Delta_v$;
4. if $w\mathcal{R}v$ and $\varphi \rightarrow \psi \in \Gamma_w$ then $\psi \in \Gamma_v$ or $\varphi \in \Delta_v$;
5. if $v\mathcal{R}w$ and $\varphi \prec \psi \in \Delta_w$ then $\psi \in \Gamma_v$ or $\varphi \in \Delta_{w'}$;
6. if $w\mathcal{R}v$ and $\varphi \in \Gamma_w$ then $\varphi \in \Gamma_v$;
7. if $v\mathcal{R}w$ and $\varphi \in \Delta_w$ then $\varphi \in \Delta_v$.

We now show that given a model graph, we can use it to construct a **BiInt** model.

Lemma 12. If there exists a model graph $\langle \mathcal{W}, \mathcal{R} \rangle$ for $\Gamma \vdash \Delta$, then there exists a **BiInt** model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$ such that for some $w_0 \in \mathcal{W}$, we have $w_0 \models \Gamma$ and $w_0 \models \Delta$. We call \mathcal{M} the counter-model for $\Gamma \Vdash_{\text{BiInt}} \Delta$.

Proof. Since we already have a **BiInt** frame $\langle \mathcal{W}, \mathcal{R} \rangle$, we need to define a valuation ϑ in order to construct a **BiInt** model $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \vartheta \rangle$:

1. For every world $w \in \mathcal{W}$ and every atom $p \in \Gamma_w$, let $\vartheta(w, p) = \text{true}$.
2. For every world $w \in \mathcal{W}$ and every atom $q \in \Delta_w$, let $\vartheta(w, q) = \text{false}$.

Then properties 6 and 7 of Definition 25 ensure persistence and reverse persistence respectively.

We now need to show that for every world $w \in \mathcal{W}$, we have $w \models \Gamma_w$ and $w \models \Delta_w$; we can do this by simple induction on the length of the formulae in $\Gamma \vdash_w \Delta$.

Now let w_0 be the world in the model graph such that $\Gamma \subseteq \Gamma_{w_0}$ and $\Delta \subseteq \Delta_{w_0}$. Since our proof by induction has shown that for every world $w \in \mathcal{W}$, we have $w \models \Gamma_w$ and $w \models \Delta_w$, then in particular, we have that $w_0 \models \Gamma_{w_0}$ and $w_0 \models \Delta_{w_0}$. Then, since we have that $\Gamma \subseteq \Gamma_{w_0}$ and $\Delta \subseteq \Delta_{w_0}$, we also have $w_0 \models \Gamma$ and $w_0 \models \Delta$.

5.4 Main Completeness Proof

We now show how to construct a model graph for $\Gamma \vdash \Delta$ from a consistent $\Gamma \vdash \Delta$. Recall from Remark 2 that we need to keep a number of independent versions of worlds because of the choices arising due to disjunctive non-determinism. We do this by storing one or more independent connected-components $\langle \mathcal{W}_1, \mathcal{R}_1 \rangle, \dots, \langle \mathcal{W}_n, \mathcal{R}_n \rangle$ in the constructed model graph $\langle \mathcal{W}, \mathcal{R} \rangle$, and the indices (sorts) of worlds and

Procedure MGC

Input: sequent $\Gamma \vdash \Delta$

Output: model graph $\langle \mathcal{W}^f, \mathcal{R}^f \rangle$, variables \mathcal{S}^f and \mathcal{P}^f

1. Let $\zeta = \{\alpha_1, \dots, \alpha_n\}$ be the result of saturating $\Gamma \vdash \Delta$ using Lemma 11;
2. For each $\alpha_i \in \zeta$ do
 - (a) Let $\langle \mathcal{W}_i, \mathcal{R}_i \rangle = \langle \{\alpha_i\}, \{(\alpha_i, \alpha_i)\} \rangle$; let $recompute := false$;
 - (b) For each non-blocked $\varphi \rightarrow \psi \in \Delta_{\alpha_i}$ and while $recompute = false$ do
 - i. Apply (\rightarrow_R) to $\varphi \rightarrow \psi$ and obtain a left premise $\pi_1 = \Gamma_{\alpha_i}, \varphi \vdash \psi$;
 - ii. Let $\langle \mathcal{W}, \mathcal{R} \rangle, \mathcal{S}, \mathcal{P} := MGC(\pi_1)$;
 - iii. If $\exists \Pi_j \in \mathcal{P}. \Pi_j \subseteq \Delta_{\alpha_i}$ then
 - A. Let $u_j \in \mathcal{W}_j$ be the root of the connected component \mathcal{W}_j from \mathcal{W} ;
 - B. Let $G = \langle \mathcal{W}_j, \mathcal{R}_j \rangle[j := i]$; add G to $\langle \mathcal{W}_i, \mathcal{R}_i \rangle$, and put $\alpha_i \mathcal{R}_i u_i$.
 - iv. else
 - A. Let $\langle \mathcal{W}_i, \mathcal{R}_i \rangle = \langle \epsilon, \epsilon \rangle$; let $recompute := true$;
 - B. Invoke the right premise of (\rightarrow_R) to obtain $\pi_2 = \Gamma_{\alpha_i} \vdash \Delta_{\alpha_i}, \bigwedge \mathcal{P}$;
 - C. Apply (\bigwedge_R) to π_2 to obtain $m \geq 1$ non-derivable premises $\gamma_1, \dots, \gamma_m$;
 - D. For each $\gamma_k, 1 \leq k \leq m$, let $\langle \mathcal{W}_k, \mathcal{R}_k \rangle, \mathcal{S}_k, \mathcal{P}_k := MGC(\gamma_k)$;
 - E. Let $\langle \mathcal{W}_i, \mathcal{R}_i \rangle := \langle \bigcup \mathcal{W}_k, \bigcup \mathcal{R}_k \rangle$, and $\mathcal{S}_i := \bigcup \mathcal{S}_{\gamma_k}$ and $\mathcal{P}_i := \bigcup \mathcal{P}_{\gamma_k}$;
 - (c) For each non-blocked $\varphi \prec \psi \in \Gamma_{\alpha_i}$ and while $recompute = false$ do
 - i. Perform a symmetric procedure to Steps 2(b)i to 2(b)ivE.
 - (d) If $recompute = false$ then let $\mathcal{S}_i := \{\Gamma_{\alpha_i}\}$ and $\mathcal{P}_i := \{\Delta_{\alpha_i}\}$.
3. Return $\langle \bigcup \mathcal{W}_i, \bigcup \mathcal{R}_i \rangle, \bigcup \mathcal{S}_i, \bigcup \mathcal{P}_i$

Fig. 4. Model Graph Construction Procedure

relations tell us the connected-component of the graph to which they belong. We write $\langle \mathcal{W}_j, \mathcal{R}_j \rangle[j := i]$ to relabel the connected component $\langle \mathcal{W}_j, \mathcal{R}_j \rangle$ with sort j to a connected component $\langle \mathcal{W}_i, \mathcal{R}_i \rangle$ with sort i . Similarly, we also label each member of the variables \mathcal{P} and \mathcal{S} , so we can later extract the member with sort i , corresponding to the component of $\langle \mathcal{W}, \mathcal{R} \rangle$ with sort i . We write \mathcal{R} -neighbour to mean \mathcal{R} -predecessor or \mathcal{R} -successor.

Our algorithm in Fig. 4 starts by saturating the root world to obtain one or more saturated “states”. For each “state” α_i , it recursively creates all the \mathcal{R} -neighbours and saturates them, and so on. If during the construction of any \mathcal{R} -neighbour, new information is returned from the higher sequents (Step 2(b)iv), then we delete the entire subtree (connected component of sort i) rooted at α_i , and recreate α_i using the new information (Step 2(b)ivB). This re-creates all the \mathcal{R} -neighbours of α_i . Otherwise, if none of the \mathcal{R} -neighbours of α_i return any new information, or there are no \mathcal{R} -neighbours for α_i , then Step 2d instantiates the variables and returns from the recursion. In the latter case, the “state” α_i already has all the required information it can possibly receive from any \mathcal{R} -neighbours, thus α_i is final. Note the duality: new information *from* a single \mathcal{R} -neighbour means that all of the members of a variable were new, while new information *at* a “state” α_i means that some \mathcal{R} -neighbour returned new information.

When we return from *MGC*, we form the union of the components of the model graph and the variables from the different “states”, so that the caller of *MGC* can extract the appropriate component at Step 2(b)iiiA.

Remark 3. Note that while the counter-model construction procedure keeps the whole counter-model in memory, this procedure is only used to prove the completeness of **GBiInt**. Our procedure for checking the validity of **BiInt** formulae (Fig. 3) does not need the whole counter-model, and explores one branch at a time, as is usual for sequent/tableaux calculi.

Theorem 3 (Completeness). *GBiInt is complete: if $\Gamma \vdash \Delta$ is not derivable, then there exists a counter-model for $\Gamma \Vdash_{\text{BiInt}} \Delta$.*

Proof. Suppose $\Gamma \vdash \Delta$ is not derivable, then by Corollary 4 we have that $\Gamma \vdash \Delta$ is consistent. We construct a model graph for $\Gamma \vdash \Delta$ using the procedure given in Figure 4, and obtain $\langle \mathcal{W}^f, \mathcal{R}^f \rangle$. We let $\langle \mathcal{W}, \mathcal{R} \rangle$ be any connected component of $\langle \mathcal{W}^f, \mathcal{R}^f \rangle$. We now show that $\langle \mathcal{W}, \mathcal{R} \rangle$ satisfies the properties of a model graph from Definition 25:

1. $\Gamma \subseteq \Gamma_{w_0}$ and $\Delta \subseteq \Delta_{w_0}$ for some $w_0 \in \mathcal{W}$: This holds because w_0 is one of the saturated sequents obtained from $\Gamma \vdash \Delta$. Moreover, if we delete the original w_0 at Step 2(b)ivA, a final version of w_0 is created at Step 2(b)iiiB which is never deleted.
2. if $\varphi \rightarrow \psi \in \Delta_w$ then $\exists v \in \mathcal{W}$ with $w\mathcal{R}v$ and $\varphi \in \Gamma_v$ and $\psi \in \Delta_v$: This holds because we have either created v using (\rightarrow_R) at Step 2(b)iiiB, or had w fulfill the role of this successor by reflexivity if (\rightarrow_R) was blocked.
3. if $\varphi \prec \psi \in \Gamma_w$ then there exists some $v \in \mathcal{W}$ with $v\mathcal{R}w$ and $\varphi \in \Gamma_v$ and $\psi \in \Delta_v$:

By symmetry with property 2.

4. if $w\mathcal{R}v$ and $\varphi \rightarrow \psi \in \Gamma_w$ then $\psi \in \Gamma_v$ or $\varphi \in \Delta_v$: In our construction, there are three ways of obtaining $w\mathcal{R}v$, so we need to show that for each case, the property holds. We first show that $\varphi \rightarrow \psi \in \Gamma_v$:
 - (a) v was created by applying (\rightarrow_R) to w on some $\alpha \rightarrow \beta \in \Delta_w$. Then Γ_v also contains $\varphi \rightarrow \psi$.
 - (b) w was created by applying (\prec_L) to some $\alpha \prec \beta \in \Gamma_v$. Then, when the final version of Γ_v was created, $\varphi \rightarrow \psi \in \Gamma_w$ was added to the \mathcal{S} variable at Step 2d. There are two cases:
 - The right premise π_2 of (\prec_L) was invoked at v . Then \mathcal{S} was added to π_2 at v by the symmetric process to Step 2(b)ivB. Thus the updated Γ_v also contains $\varphi \rightarrow \psi$.
 - The right premise of (\prec_L) was not invoked at v . This means that $\exists \Sigma_j \in \mathcal{S}. \Sigma_j \subseteq \Gamma_v$, and the j -th version of v 's predecessor w is chosen at the symmetric process to Step 2(b)iiiA. But since Step 2d at w assigns $\Sigma_j := \Gamma_w$, then we have $\Gamma_w \subseteq \Gamma_v$ and thus $\varphi \rightarrow \psi \in \Gamma_v$.
 - (c) $v = w$, and $w\mathcal{R}w$ by reflexivity. Then $\Gamma_v = \Gamma_w$, so $\varphi \rightarrow \psi \in \Gamma_v$.
- In all cases, saturation for v will then ensure that $\psi \in \Gamma_v$ or $\varphi \in \Delta_v$.

5. if $v\mathcal{R}w$ and $\varphi \prec \psi \in \Delta_w$ then $\psi \in \Gamma_v$ or $\varphi \in \Delta_v$:

By symmetry with property 4.

6. if $w\mathcal{R}v$ and $\varphi \in \Gamma_w$ then $\varphi \in \Gamma_v$:

By similar argument to property 4.

7. if $v\mathcal{R}w$ and $\varphi \in \Delta_w$ then $\varphi \in \Delta_v$:

By symmetry with property 6.

We can obtain a counter-model for $\Gamma \Vdash_{\text{BiInt}} \Delta$ from $\langle \mathcal{W}, \mathcal{R} \rangle$ via Lemma 12.

Definition 26. A *di-tree* is a directed graph such that if the direction of the edges is ignored, it is a tree.

Theorem 4. Every falsifiable **BiInt** sequent can be falsified by a model whose frame is a di-tree, consisting of reflexive points.

Proof. From Lemmas 5 and 6, we know that the construction of new successors for $\varphi \rightarrow \psi$ and predecessors for $\varphi \prec \psi$ stops when either there are no rejected $\varphi \rightarrow \psi$ -formulae or forced $\varphi \prec \psi$ -formulae in the current world, or the current world already forces φ and rejects ψ . In the latter case, the world itself fulfills the role of the successor or predecessor by reflexivity, and no new successors or predecessors are created.

The reason we are able to avoid proper cycles is the persistence and reverse persistence properties of **BiInt**, used in the (\rightarrow_R^I) and (\prec_L^I) rules.

Consider the \rightarrow case. Every time some $\varphi \rightarrow \psi$ appears on the RHS of a sequent $\Gamma \vdash \Delta, \varphi \rightarrow \psi$, we first add ψ to the RHS to obtain $\Gamma \vdash \Delta, \varphi \rightarrow \psi, \psi$ using the (\rightarrow_R^I) rule, since by reverse persistence the current world must reject everything that some successor world rejects. Now that ψ is on the RHS, we need to apply the (\rightarrow_R) rule to create the $\varphi \rightarrow \psi$ -successor $\Gamma, \varphi \vdash \psi$ only if φ is not already on the LHS. For if $\varphi \in LHS$, then the successor $\Gamma \vdash \psi$ that fulfills $\varphi \rightarrow \psi$ can be the current world itself. So there is no point creating it explicitly.

Corollary 5. **BiInt** is characterised by finite rooted reflexive and transitive di-trees of reflexive points.

6 Conclusions and Future Work

Our cut-free calculus for **BiInt** enjoys terminating backward proof-search and is sound and complete w.r.t Kripke semantics. A simple Java implementation of **GBiInt** is available at <http://users.rsise.anu.edu.au/~linda>. The next step is to add a cut rule to **GBiInt**, and prove cut elimination syntactically. We are also extending our work to the modal logic **S5**, and the tense logic **Kt.S4**. Our approach of existential branching and inter-premise communication bears some similarities to hypersequents of Pottinger and Avron [1]. It would be interesting to investigate this correspondence further. From an automated deduction perspective, **GBiInt** is the first step towards an efficient decision procedure for **BiInt**. The next task is to analyse the computational complexity of **GBiInt** and investigate which of the traditional optimisations for tableaux systems are still applicable in the intuitionistic case.

We would like to thank the anonymous reviewers for their suggestions.

References

1. Arnon Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In Wilfrid Hodges, Martin Hyland, Charles Steinhorn, and John Truss, editors, *Logic: from foundations to applications. Proc. Logic Colloquium, Keele, UK, 1993*, pages 1–32. Oxford University Press, New York, 1996.
2. T. Crolard. Subtractive logic. *Theor. Comp. Sci.*, 254(1–2):151–185, March 2001.
3. T. Crolard. A formulae-as-types interpretation of Subtractive Logic. *Journal of Logic and Computation*, 14(4):529–570, August 2004.
4. J. Czermak. A remark on Gentzen’s calculus of sequents. *Notre Dame Journal of Formal Logic*, 18(3):471–474, 1977.
5. A. Dragalin. *Mathematical Intuitionism: Introduction to Proof Theory*, volume 68 of *Translations of Mathematical Monographs*. Cambridge Univ. Press, 1988.
6. R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, September 1992.
7. J. H. Gallier. *Logic for Computer Science, Foundations of Automated Theorem Proving*. Computer Science and Technology Series. Harper & Row, 1986.
8. G. Gentzen. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1935. English translation in [19].
9. R. Goré. Tableau methods for modal and temporal logics. In D’Agostino et al., editor, *Handbook of Tableau Methods*, pages 297–396. Kluwer, 1999.
10. A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Analytic Tableaux and Related Methods*, volume 1071 of *LNAI*, pages 210–225, 1996.
11. I. Horrocks, U. Sattler, and S. Tobies. A PSpace-algorithm for deciding $ALCNI_{R+}$ -satisfiability. Technical Report LTCS-98-08, LuFG Theoretical Computer Science, RWTH Aachen, 1998.
12. J. M. Howe. *Proof search issues in some non-classical logics*. PhD thesis, University of St Andrews, 1998.
13. J. C. C. McKinsey and A. Tarski. On closed elements in closure algebras. *Annals of Mathematics*, 47(1):122–162, 1946.
14. D. Nelson. Constructible falsity. *Journal of Symbolic Logic*, 14(2):16–26, 1949.
15. C. Rauszer. A formalization of the propositional calculus of H-B logic. *Studia Logica*, 33:23–34, 1974.
16. C. Rauszer. An algebraic and Kripke-style approach to a certain extension of intuitionistic logic. *Dissertationes Mathematicae*, 168, 1980. Institute of Mathematics, Polish Academy of Sciences.
17. S. Schwendimann. A new one-pass tableau calculus for PLTL. In *Analytic Tableaux and Related Methods*, volume 1397 of *LNAI*, pages 277–292, 1998.
18. V. Švejdar. On sequent calculi for intuitionistic propositional logic. *Commentationes Mathematicae Universitatis Carolinae*, 47(1):159–173, 2006.
19. M. E. Szabo, editor. *The Collected Papers of Gerhard Gentzen*. Studies in Logic and the foundations of Mathematics. North-Holland, Amsterdam, 1969.
20. I. Urbas. Dual-intuitionistic logic. *Notre Dame Journal of Formal Logic*, 37(3):440–451, Summer 1996.
21. T. Uustalu. Personal communication. via email, 2004.
22. T. Uustalu. Personal communication. via email, 2006.
23. T. Uustalu and L. Pinto. Days in logic ’06 conference abstract. Online at <http://www.mat.uc.pt/~kahle/dl06/tarmo-uustalu.pdf>, accessed on 27th October 2006, 2006.

24. F. Wolter. On logics with coimplication. *Journal of Philosophical Logic*, 27(4):353–387, 1998.